

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл. ім'я, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

на тему: «Інформаційна система підтримки діяльності
розкрійного цеху меблевого виробництва»

Виконав:

студент IV курсу, групи ІС-63

Мосьняк Олег Ігорович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., Жураковська Оксана Сергіївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант
з графічної
документації**

ст.викл., Проскура Світлана Леонідівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

доц. каф. ТК, к.т.н., доц. Лісовиченко Олег Іванович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“ ” 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Мосьпаку Олегу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна система підтримки діяльності
розкрийного цеху меблевого виробництва»

керівник проєкту Жураковська Оксана Сергіївна, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного
середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі,
обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до
технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань
програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна варіантів використання*

2. *Схема структурна діяльності*

3. *Схема бази даних*

4. *Схема структурна класів програмного забезпечення*

5. *Схема структурна послідовності*

6. *Схема структурна компонентів*

7. *Креслення вигляду екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>14.04.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>18.04.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>20.04.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>22.04.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>26.04.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>28.04.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>30.04.2020</i>	
8.	<i>Налагодження програми</i>	<i>11.05.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>12.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>14.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Олег МОСЬПАК

Керівник

Оксана ЖУРАКОВСЬКА

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Інформаційна система підтримки діяльності розкрійного цеху
меблевого виробництва

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 30 рисунків, 23 таблиці, 1 додаток, 22 джерело.

Дипломний проєкт присвячений розробці інформаційної системи підтримки діяльності розкрійного цеху меблевого виробництва.

Метою роботи є підвищення ефективності виробничого процесу розкрійного цеху меблевого виробництва.

У розділі інформаційного забезпечення описано вхідні дані, вихідні дані та структуру бази даних.

Розділ математичного забезпечення присвячений змістовній постановці задачі, математичній поставці задачі розкрою, яка є базовою задачею для діяльності розкрійного цеху меблевого виробництва, обґрунтуванню та описанню методів розв'язання поставленої задачі.

Розділ програмного забезпечення присвячений засобам розробки, вимогам до технічного забезпечення, архітектурі програмного забезпечення та звітам.

У технологічному розділі описано керівництво користувача та випробування програмного продукту.

**ІНФОРМАЦІЙНА СИСТЕМА, ПІДТРИМКА ДІЯЛЬНОСТІ,
РОЗКРІЙНИХ ЦЕХ, МЕБЛЕВЕ ВИРОБНИЦТВО, ЗАДАЧА
РОЗКРОЮ.**

					<i>ДП 6316.00.000 ПЗ</i>		
		<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Мосьняк О.І.</i>			<i>Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва</i>		
<i>Перевірив.</i>		<i>Жураковська О.С.</i>					
<i>Н. кон.</i>		<i>Проскура С.Л.</i>					
<i>Затв.</i>		<i>Павлов О.А.</i>					
					<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>
						2	85
					<i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63</i>		

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of five sections, contains 30 drawings, 23 tables, 1 application, 22 sources.

The diploma project is devoted to the development of an information system for furniture industry to optimize cutting shop processes.

The purpose of the work is to increase the efficiency of the production process of the cutting shop of the furniture production.

The information support section describes the input data, the output data and the database structure.

The section of mathematical support is devoted to the meaningful formulation of the problem, mathematical delivery of the cutting problem, which is the basic task for the cutting shop of furniture production, substantiation and description of methods for solving the problem.

The software section focuses on development tools, hardware requirements, software architecture and reports.

The technology section describes the user manual and the software test.

INFORMATION SYSTEM, SUPPORT OF ACTIVITIES, CUTTING SHOP, FURNITURE PRODUCTION, CUTTING TASK.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ЗМІСТ

ВСТУП	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	8
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	8
1.1.1 Опис процесу діяльності	9
1.1.2 Опис функціональної моделі	9
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	10
1.3 ПОСТАНОВКА ЗАДАЧІ	11
1.3.1 Призначення розробки	11
1.3.2 Цілі та задачі розробки	12
Висновок до розділу	12
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	14
2.1 ВХІДНІ ДАНІ	14
2.2 ВИХІДНІ ДАНІ	16
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	17
Висновок до розділу	19
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	21
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	21
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	21
3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ	23
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ	24
Висновок до розділу	32
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	34
4.1 ЗАСОБИ РОЗРОБКИ	34
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	37
4.2.1 Загальні вимоги	37
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	37
4.3.1 Діаграма класів	37
4.3.2 Діаграма послідовності	38
4.3.3 Діаграма компонентів	39
4.3.4 Специфікація функцій	39

4.4	ОПИС ЗВІТІВ	40
	Висновок до розділу	42
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	43
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА	43
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	60
5.2.1	Мета випробувань	60
5.2.2	Загальні положення	60
5.2.3	Результати випробувань	60
	Висновок до розділу	68
	ЗАГАЛЬНІ ВИСНОВКИ	70
	ПЕРЕЛІК ПОСИЛАНЬ	72
	ДОДАТОК А	75

ВСТУП

Для будь-якого застарілого підприємства, актуальність використання інформаційних технологій є надважливою складовою, бо воно зазнає збитки.

На теперішній час не вигідно просто взяти якесь підприємство та придбати декілька нових станків чи найняти ще більшу кількість людей для підвищення ефективності. Це не вигідно, тому що, якщо конкурент збудує нове підприємство, яке повністю буде автоматизоване та працювати буде мінімальна кількість людей – то людина зі своїм старим підприємством, яке має звичайні станки, велику кількість робітників, яким необхідно платити зарплату не зможе конкурувати, через те що конкурент буде платити менше грошей робітникам та матиме нові станки, які працюють швидше та мають можливість автоматично робити справу.

Подивимось на сфери, які ще не використовують інформаційні технології за для підвищення ефективності виробництва. Це, наприклад, меблеве виробництво, скляне виробництво, виробництво запчастин для літаків. Відносно виробництва запчастин для літаків, можна сказати, що Україна не погано виробляє запчастини літаків, але останній час ефективність спала, та частково це сталося через те, що до сих пір підприємство наймає робітників, які ще працюють на ручних станках, але звісно поступово купується станки, які працюють автоматично. Складне та меблеве виробництво схоже за структурою, бо в обох необхідно розкroїти матеріал. В меблевому виробництві використовується розкрій деревини, а в скляному виробництві використовується розкрій скла. В Україні гірша ситуація по автоматизації цих підприємств, ніж закордоном. Вибір пав на меблеве виробництво, бо в майбутньому, можливо, автором буде створено меблеве виробництво, а ефективність меблевого виробництва (тобто збільшення прибутку, зменшення часу на розкрій) можна підвищити завдяки купівлі нових станків чи автоматизації розкрою. Саме автоматизація розкрою є найпростішим та

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

найдешевшим шляхом підвищення ефективності, через це цей шлях було обрано в дипломному проєкті.

Отже, необхідно розробити інформаційну систему підтримки діяльності розкрійного цеху меблевого виробництва, що дозволить підвищити ефективність виробництва та зменшити час на розкрій.

Дипломний проєкт присвячений розробці інформаційної системи підтримки діяльності розкрійного цеху меблевого виробництва.

Практичне значення одержаних результатів. Впровадження системи дасть економічний ефект, тобто підвищення прибутку.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Якщо розглядати розкрійний цех меблевого виробництва як об'єкт управління, то характерними особливостями є те, що однією з основних задач процесу виробництва, вирішення якої впливає на затрати ресурсів та розмір прибутку від виробництва, є задача розкрою матеріалів. Задача повинна відповідати наступним вимогам: мінімізація відходів та часу на розкрій. Саме для цього необхідно створення програмного продукту, який вирішить проблему меблевого виробництва, тому розробка даної інформаційної системи є актуальною.

Так як більшість меблевих виробництв були збудовані десятиліттями тому, вони вважаються застарілими та не використовують сучасні технології у виробництві: використання програмного забезпечення для ефективного розкрою, використання баз даних. Це потенційно може призвести до спаду доходів компанії, як це відбувається часто з державними підприємствами, бо вони були збудовані давно, а потім вони банкрутують та зачиняють виробництво. Щоб цього запобігти, потрібно автоматизувати процеси, використовувати інформаційні системи та технології, які допоможуть зробити процес виробництва більш ефективним з точки зору раціонального використання обмежених ресурсів та часу, а також дозволить підвищити прибутковість виробництва.

Найбільша складова в підвищенні ефективності та прибутковості існуючого виробництва припадає на розробку та впровадження інформаційної системи підтримки процесу меблевого виробництва, однією із основних задач якого є управління процесом розкрою розкрійного цеху.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1.1.1 Опис процесу діяльності

Менеджеру меблевого виробництва приходить заказ від замовника на розкрій. Менеджер заносить дані у базу даних. Після цього йде процес формування плану розкрою. Результати розкрою заносяться в базу даних. Менеджер формує звіт та повідомляє замовника о результатах. При необхідності менеджер формує завдання на розкрій.

Перед тим як проектувати діаграму бізнес-процесу потрібно визначити актора. Актором застосунку є користувач, тобто менеджер розкрійного цеху, який і буде користуватися програмним продуктом.

Схема структурна діяльності наведена в графічному матеріалі.

В застосунку менеджер може обрати наступне:

- «Вибір варіанту розкрою в діалоговому режимі», менеджер може ввести дані самостійно для формування плану розкрою;
- «Вибір варіанту розкрою на основі даних замовників», менеджер може додавати, редагувати, видаляти дані замовників у базі даних та обирати для якого замовника робити формування плану розкрою.

1.1.2 Опис функціональної моделі

Функціональна модель системи представлена у вигляді моделі варіантів використання. Схема структурна варіантів використання наведена в графічному матеріалі.

Актором системи є користувач, тобто менеджер розкрійного цеху. Функції, які буде виконувати актор, наведені в таблиці 1.1. В таблиці є опис актора, опис варіантів використання та опис функцій варіантів використання.

Таблиця 1.1 – Функції варіантів використання

Актор	Варіант використання	Опис функцій варіантів використання
Менеджер	Формування плану розкрою	Менеджер має змогу сформувати план розкрою
	Ведення замовників	Менеджер має змогу додавати, видаляти, редагувати замовника
	Ведення заготовок	Менеджер має змогу додавати, видаляти, редагувати заготовки
	Перегляд результатів	Менеджер має змогу переглянути результати в таблицях та видалити, якщо необхідно. Менеджер має змогу переглянути звіт
	Аналіз результатів	Менеджер має змогу порівняти результати та додати до таблиць кращий результат
	Формування звіту	Менеджер має змогу сформувати звіт для замовника щойно програма видала результат
	Формування завдання на розкрій	Менеджер має змогу сформувати завдання на розкрій щойно програма видала результат

1.2 Огляд наявних аналогів

Огляд аналогічних продуктів показав, що аналогічні застосування не дають можливості вирішення задачі розкрою матеріалу за допомогою методу двоетапного розкрою без можливості повороту дошки та не використовують

алгоритми для максимізації прибутку завдяки розкрою, тому можна стверджувати, що для даної системи конкурентів немає. Проте обрано дві програми для яких зроблений короткий критичний аналіз, наведений в таблиці 1.2.

Таблиця 1.2 – Критичний аналіз аналогів

Програма	Переваги	Недоліки
«Мастер 2»	<ol style="list-style-type: none"> 1. Зручний графічний інтерфейс. 2. Є можливість вести підприємство. 3. Наявне використання багатьма користувачами. 	<ol style="list-style-type: none"> 1. Платний функціонал. 2. Не має алгоритмів для ефективного розкрою. 3. Потребує багато ресурсів комп'ютера, високі вимоги до оперативної пам'яті.
«Астра Раскрой»	<ol style="list-style-type: none"> 1. Розкрій на ЧПУ-станках. 2. Працює з склом, пластмасою, деревиною. 3. Працює з програмою «1С». 	<ol style="list-style-type: none"> 1. Платний функціонал. 2. Необхідний інтернет для роботи. 3. Дуже багато функцій, які не використовуються при організації виробничого процесу розкрійного цеху.

Отже, необхідно розробити систему, яка б дозволила підвищити ефективність роботи розкрійного цеху за рахунок вирішення задачі розкрою матеріалів за умови раціонального використання наявних ресурсів та максимізації прибутку.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням розробки є підтримка діяльності розкрійного цеху меблевого виробництва.

1.3.2 Цілі та задачі розробки

Метою є підвищення ефективності виробничого процесу розкрійного цеху меблевого виробництва.

Щоб досягти поставлену мету, потрібно вирішити наступні задачі:

- проведення аналізу алгоритмів, які підходять до поставленої задачі;
- розробка алгоритмів вирішення задачі розкрою для максимізації прибутку;
- ведення замовників;
- ведення заготовок замовників;
- формування планів розкрою;
- створення програмного застосування;
- формування звітів;
- формування завдання на розкрій.

Висновок до розділу

У розділі «Загальні положення» розглянуто предметне середовище, аналоги існуючих систем, описано функціональну модель, процес діяльності та призначення розробки. Також сформована мета та задачі, які допомагають досягти поставленої мети.

На основі аналізу предметного середовища, було описано наявні проблеми на теперішній час будь-якого виробництва, розглянуто структуру меблевого виробництва та визначено задачі, вирішення яких дозволить підвищити ефективність процесу виробництва та здійснити планування та організацію цього процесу за критерієм максимізації прибутку.

Наведено схему структурну діяльності, яка поділена на три сектори: актор, вхідна/вихідна інформація та діяльність. Актором виступає менеджер.

Розроблено функціональну модель системи. Описано акторів системи, їх варіанти використання та функціональні вимоги. Наведено схему структурну варіантів використання.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Проаналізовано існуючі аналоги для програмного забезпечення, яке використано в даному дипломному проєкті. Проведений аналіз показав доцільність розробки системи, яка б дозволила мінімізувати наявні недоліки і досягти поставленої мети в підтримці процесу управління меблевим виробництвом.

Виконано постановку задачі: описано призначення розробки, сформовано мету, яка їй відповідає суті цієї роботи та поставлені задачі, які приведуть до мети: підвищення ефективності роботи розкрійного цеху меблевого виробництва.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

В інформаційній системі є два варіанти вхідних даних:

- вхідні дані, які менеджер самостійно вводить для формування плану розкрою;
- вхідні дані, які менеджер обирає в базі даних для якого замовника сформувати план розкрою.

Для того, щоб програма працювала вірно, на всі варіанти вхідних даних накладаються обмеження, наведені в таблиці 2.1.

Таблиця 2.1 – Обмеження для вхідних даних

Вхідні дані	Тип даних	Діапазон даних
Довжина головної дошки	int	[0...1000000]
Ширина головної дошки	int	[0...1000000]
Кількість заготовок	int	[0...1000000]
Довжина заготовки	int	[0...1000000]
Ширина заготовки	int	[0...1000000]
Прибутковість заготовки	int	[0...1000000]

При роботі інформаційної системи, коли обрано перший варіант, дані вводяться самостійно для формування плану розкрою в призначені поля:

- довжина головної дошки;
- ширина головної дошки;
- кількість заготовок;
- довжини заготовок;
- ширини заготовок;
- прибутковість заготовок.

Для полів, де необхідно вводити вектори заготовок, дані вводяться через пробіл.

При роботі інформаційної системи, коли обрано другий варіант, менеджер обирає замовника, тобто вводить Id замовника у відповідне поле. Дані зчитуються з відповідних стовпців таблиць для формування плану розкрою в ті самі призначені поля, в які можна ввести дані самостійно. Назви стовпців наведені в таблиці 2.2.

Таблиця 2.2 – Пояснення назв стовпців для вхідних даних

Назва таблиці	Назва стовпця	Пояснення
«Customers» - таблиця із даними замовників	Id	Ідентифікатор замовника
	Length_Main	Довжина головної дошки
	Width_Main	Ширина головної дошки
	Count_Type_Bars	Кількість заготовок
«Bars» - таблиця із даними заготовок замовників	Customer_Id	Ідентифікатор замовника
	Bar	Тип заготовки
	Length	Довжина заготовки
	Width	Ширина заготовки
	Profit	Прибутковість заготовки

Також для обох варіантів є логічні обмеження вхідних даних, при яких розроблені алгоритми не будуть працювати та повідомлять про це менеджера:

- не введені вхідні дані;
- від’ємні значення чи 0 не може бути використано;
- кількість заготовок не відповідає кількості заготовок введених у вектори;
- максимальна довжина заготовки не може бути більша за довжину головної дошки;
- максимальна ширина заготовки не може бути більша за ширину головної дошки;
- не може бути декілька однакових за шириною та довжиною заготовок;
- відношення ціни до площі заготовки для декількох заготовок не може бути однаковим.

2.2 Вихідні дані

Результатами роботи інформаційної системи є вихідні дані, які можуть бути записані двома варіантами:

- результати записані у відповідні поля вихідних даних;
- результати записані у відповідні поля вихідних даних та у поля результуючих таблиць бази даних.

Для обох варіантів вихідні дані записуються у поля:

- прибуток – це прибуток отриманий роботою обраного алгоритму;
- вектор кількості використаних заготовок – це вектор кількості використаних заготовок для досягнення максимального прибутку, завдяки обраного алгоритму.

Також для другого варіанту в таблиці результатів бази даних записуються вихідні дані у стовпці, назви яких наведені в таблиці 2.3.

Таблиця 2.3 – Пояснення назв стовпців для вихідних даних

Назва таблиці	Назва стовпця	Пояснення
«Result_Bars» - таблиця із кількістю використаних заготовок для максимального прибутку, який вважається кінцевим результатом	Customer_Id	Ідентифікатор замовника
	Algorithm	Обраний алгоритм для розрахунку
	Bar	Тип заготовки
	Count_Bars	Кількість використаних заготовок

Продовження таблиці 2.3.

Назва таблиці	Назва стовпця	Пояснення
«Result_Income» - таблиця із загальним прибутком та ефективністю роботи алгоритмів	Customer_Id	Ідентифікатор замовника
	Algorithm	Обраний алгоритм для розрахунку
	Total_Income	Загальний прибуток

2.3 Опис структури бази даних

Для даного дипломного проєкту використано СУБД SQL Server. База даних складається з чотирьох таблиць:

- «Customers»;
- «Bars»;
- «Result_Bars»;
- «Result_Income».

Таблиця «Customers» з'єднана з іншими таблицями завдяки Id замовника. Схема структурна бази даних наведена в графічному матеріалі.

Головна таблиця «Customers» – таблиця для вхідних даних, яка має інформацію про замовників, а також параметри для головної дошки та кількості заготовок.

Таблиця «Bars» – таблиця для вхідних даних, яка має інформацію про заготовки замовників.

Таблиця «Result_Bars» – таблиця для вихідних даних, яка має інформацію про кількість використаних заготовок замовників.

Таблиця «Result_Income» – таблиця для вихідних даних, яка має інформацію про загальний прибуток замовника та ефективність використання алгоритмів.

Структура таблиць бази даних наведено в таблиці 2.4.

Таблиця 2.4 – Структура таблиць бази даних

Таблиця	Назва стовпця	Тип даних	Пояснення
«Customers»	Id	int	Ідентифікатор замовника
	Name	nvarchar(50)	Ім'я замовника
	Surname	nvarchar(50)	Прізвище замовника
	Phone	int	Номер телефону замовника
	Length_Main	int	Довжина головної дошки замовника
	Width_Main	int	Ширина головної дошки замовника
	Count_Type_Bars	int	Кількість заготовок замовника
«Bars»	Id	int	Ідентифікатор заготовки замовника
	Customer_Id	int	Ідентифікатор замовника
	Bar	int	Тип заготовок замовника
	Length	int	Довжина заготовки замовника
	Width	int	Ширина заготовки замовника
	Profit	Int	Прибутковість заготовки замовника

Продовження таблиці 2.4.

Таблиця	Назва стовпця	Тип даних	Пояснення
«Result_Bars»	Id	int	Ідентифікатор використаної заготовки замовника
	Customer_Id	int	Ідентифікатор замовника
	Algorithm	nvarchar(50)	Використаний алгоритм
	Bar	int	Тип заготовки замовника
	Count_Bars	int	Кількість використаних заготовок замовником
«Result_Income»	Id	int	Ідентифікатор результату замовника
	Customer_Id	int	Ідентифікатор замовника
	Algorithm	nvarchar(50)	Використаний алгоритм
	Total_Income	int	Загальний прибуток замовника

Висновок до розділу

У розділі «Інформаційне забезпечення» описано вхідні дані, вихідні дані та структуру бази даних.

Було представлено два варіанти вхідних та вихідних даних відповідно. Для вхідних даних описані обмеження типів та діапазонів даних, поля в які необхідно вводити дані чи стовпці з яких зчитуються дані. Також описані логічні обмеження даних для правильної роботи алгоритмів. Для вихідних даних описано поля в яких вони будуть відображатися та результуючі таблиці в яких зберігаються результати.

Наведено схему структурну бази даних, детально описані таблиці та зв'язки. Представлено структуру таблиць бази даних.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

В будь-якому меблевому виробництві наявний розкрійний цех. Розкрійний цех повинен використовувати програмні застосунки для підвищення ефективності виробничого процесу. Від розкрою буде залежати прибуток підприємства. При цьому, звичайно, важливо його максимізувати. Таким чином, розробка програмного продукту для розкрійного цеху меблевого виробництва є важливою. Далі опишемо постановку задачі.

Є одна головна дошка шириною W , довжиною L і n типів прямокутних заготовок шириною w_j і довжиною l_j кожна, $j = 1, \dots, n$. Кожну заготовку можна вирізати в будь-якій кількості. Відомий прибуток p_j від продажу заготовки j . Розкрій відбувається завдяки методу двоетапного розкрою без можливості повороту дошки.

Необхідно сформулювати план розкрою, для якого сумарний прибуток є максимальним.

3.2 Математична постановка задачі

Сформулюємо математичну модель даної задачі.

Змінні:

- W – це ширина головної дошки,
- L – це довжина головної дошки,
- n – це кількість заготовок,
- k – це кількість розкрійних смуг,
- w_j – це ширина j -ї заготовки, $j = 1, \dots, n$,
- l_j – це довжина j -ї заготовки, $j = 1, \dots, n$,
- p_j – це прибуток j -ї заготовки, $j = 1, \dots, n$,

- m_j – це максимальна кількість заготовок j -го типу, яка поміститься по ширині, $j = 1, \dots, n$,
- y_i – це довжина i -ї смуги, $i = 1, \dots, k$,
- x_{ij} – це кількість заготовок j -го типу, що розкрійна з i -ої смуги.

Щоб виписати умову на місткість заготовок j -го типу в i -у смугу, потрібно ввести допоміжні булеві змінні:

$$x'_{ij} = \begin{cases} 1, \text{ якщо заготовка } j - \text{го типу розкрійна з } i - \text{ї смуги.} \\ 0, \text{ інакше.} \end{cases} \quad (3.1)$$

Обмеження:

а) Необхідно знайти план розкрою, що задовольняє умови:

- 1) Загальна ширина усіх заготовок, що розкрійна з j -ї смуги, не перевищує ширини головної дошки:

$$\sum_{j=1}^n w_j x_{ij} \leq W, i = 1, \dots, k. \quad (3.2)$$

- 2) Сумарна довжина усіх смуг не перевищує довжини головної дошки:

$$\sum_{i=1}^k y_i \leq L. \quad (3.3)$$

- 3) Довжина i -ї смуги дозволяє розкроїти j -у заготовку.

$$l_j x'_{ij} = y_i, i = 1, \dots, k, j = 1, \dots, n. \quad (3.4)$$

- б) Необхідна ще група обмежень, щоб зв'язати булеві змінні x'_{ij} з цілочисельними змінними x_{ij} і гарантувати, що якщо j -а заготовка розкрійна з i -ї смуги, то можна розкроїти обмежену кількість заготовок m_j , і навпаки, не можна розкроїти j -у заготовку з i -ї смуги, якщо вона не поміщається за розмірами, тобто потрібно змодельовати дві взаємовиключних умови:

$$\begin{cases} x_{ij} = 0, \text{ якщо } x'_{ij} = 0. \\ x_{ij} \leq m_j, \text{ якщо } x'_{ij} = 1. \end{cases} \quad (3.5)$$

Це можна реалізувати за допомогою наступної нерівності:

$$x_{ij} \leq m_j x'_{ij}, i = 1, \dots, k, j = 1, \dots, n. \quad (3.6)$$

с) Остання група обмежень на значення змінних:

$$x_{ij} \leq m_j, i = 1, \dots, k, j = 1, \dots, n, \quad (3.7)$$

$$x_{ij} \geq 0, \text{ цілі}, i = 1, \dots, k, j = 1, \dots, n, \quad (3.8)$$

$$x'_{ij} \in \{0,1\}, i = 1, \dots, k, j = 1, \dots, n, \quad (3.9)$$

$$y_i \in \{1,2, \dots, L\}, i = 1, \dots, k. \quad (3.10)$$

Цільова функція:

Цільова функція – це сумарний прибуток. Необхідно сформулювати план розкрою, для якого цільова функція є максимальною.

$$\max \sum_{i=1}^k \sum_{j=1}^n p_j x_{ij} \quad (3.11)$$

3.3 Обґрунтування методу розв'язання

Метод двоетапного розкрою без можливості повороту дошки – це те саме, що і задача про двоетапний гільйотинний розкрій [7].

Задача двоетапного гільйотинного розкрою – це NP-повна задача оптимізації [1]. Задача є задачею цілочислового лінійного програмування. Наївний шлях вирішення задачі цілочислового лінійного програмування – просто ігнорувати обмеження цілочисельності на змінні x , вирішити відповідну задачу лінійного програмування (яка називається лінійним ослабленням обмежень завдання цілочислового лінійного програмування), а потім округлити компоненти рішення отриманого завдання [4]. Однак отримане рішення може виявитися не тільки не максимальним чи мінімальним, воно може виявитися навіть неприпустимим, тобто деякі обмеження можуть бути порушені.

Проаналізувавши методи вирішення задач цілочислового лінійного програмування було знайдено три варіанти: метод округлення, жадібний

алгоритм та метод гілок та меж [2].

Так як метод округлення ігнорує обмеження цілочисельності та вирішує задачу лінійного програмування, а потім просто округлює результат, в наслідок цього як раз і можна отримати недопустимий результат (ігнорувались обмеження) чи взагалі результат, який не є максимальним чи мінімальним.

Серед методів розв'язання задач цілочислового лінійного програмування виділяють клас методів гілок та меж [5]. По суті, метод гілок та меж є варіацією повного перебору з відкиданням підмножин допустимих розв'язків, що не є максимальними чи мінімальними, в залежності від цілі задачі.

Жадібний алгоритм для NP-повних задач добре підходить, бо він легкий у реалізації та швидкий і дає допустимий розв'язок, але не обов'язково цей розв'язок буде точним [10].

Отже, обрано жадібний алгоритм і метод гілок та меж для вирішення поставленої задачі. Жадібний алгоритм може використовуватися для будь-якого типу задач, але краще його використовувати для задач в яких відношення довжини головної дошки до довжин заготовок приблизно більше у 100 разів, тобто, наприклад, довжина головної дошки – два метри, довжини заготовок – два та чотири сантиметра відповідно, або коли кількість типів заготовок більша за п'ять. В усіх інших випадках можна використовувати алгоритм на основі методу гілок та меж, який дозволяє отримати точний розв'язок задачі.

3.4 Опис методів розв'язання

Спочатку опишемо розробку жадібного алгоритму для розв'язання задачі.

Ідея алгоритму: на поточному кроці розкрою обираємо таку заготовку з найкращим співвідношенням ціни і площі серед заготовок із допустимою довжиною, відрізаємо по вертикалі на головній дошці [8]. Потім заготовку із

найкращим співвідношенням, якщо її довжина така сама як і довжина відрізу, відрізаємо по ширині головної заготовки до тих пір, поки не закінчиться місце для відрізу. Якщо місце по ширині лишилось для гіршої заготовки та однакова довжина її, то відрізаємо її ширину. Далі ітераційно продовжуємо робити ці кроки.

Схема жадібного алгоритму [9] розв'язання задачі наведена в таблиці 3.1.

Таблиця 3.1 – Схема жадібного алгоритму

1	Вхід: W // Ширина головної дошки L // Довжина головної дошки l // Вектор довжин заготовок w // Вектор ширин заготовок p // Вектор прибутку від продажу заготовок n // Кількість заготовок
2	Вихід: sum //сумарний прибуток від продажу заготовок
3	v // Вектор кількості використаних заготовок
4	// Ініціалізація
5	for $i:=1$ to n do
6	$k_i = \frac{p_i}{l_i w_i}$ //Корисність виготовлення заготовки
7	end
8	$minL = \min(l)$ //Довжина найкоротшої заготовки
9	$minW = \min(w)$ //Ширина найкоротшої заготовки
10	$sort(k)$ //Сортуємо за спаданням (відповідно впорядковуються елементи векторів l і w)
11	$sum:=0, i:=1$
12	//Розкрій
13	while $L \geq minL$ AND $i \leq n$ do // Розкрій по вертикалі

Продовження таблиці 3.1.

14	// (i - індекс поточної заготовки)
15	IF ($l_i \leq L$) //Якщо довжина поточної заготовки менше залишку довжини головної дошки
16	{ $L := L - l_i$ //Відрізаємо по вертикалі
17	$j := 1$
18	while $W \geq \min W$ AND $j \leq n$ do //Розкрій по горизонталі
19	IF ($w_j \leq W$ AND $l_i = l_j$) //Якщо ширина поточної заготовки менше залишку ширини головної дошки та їх довжини однакові
20	{ $W := W - w_j$ //Відрізаємо по горизонталі
21	$sum := sum + p_j$ //Рахуємо дохід
22	$v_j := v_j + 1$ } //Рахуємо к-ть використаних заготовок типу
23	ELSE //Інакше переходимо до наступної заготовки
24	$j := j + 1$
25	END
26	}
27	ELSE //Інакше переходимо до наступної заготовки
28	$i := i + 1$
29	END

Це загальний метод, застосовний як до лінійних, так і до нелінійних задач дискретного (зокрема, цілочислового) програмування. Це комбінаторний метод, реалізований як спрямований перебір варіантів розв'язків оптимізаційних задач зазначеного типу [6].

Ідея його така. Обчислюють верхню чи нижню оцінку цільової функції на допустимій множині розв'язків, причому спосіб обчислення оцінки для кожної задачі добирають окремо.

Множину певним способом розбивають на скінченну кількість

неперетинних підмножин, на кожній з яких перевизначають оцінки цільової функції. Підмножина з кращою оцінкою перспективніша для пошуку, тому її вибирають для подальшого галуження. Інші підмножини вважають кінцевими на цьому етапі. Якщо вона має строго більшу (меншу) оцінку, ніж вибрана для галуження, то надалі її вже не розглядають. Якщо ж відмінність в оцінках для інших підмножин нестрога, то на наступних кроках можливе повернення до якоїсь кінцевої підмножини.

Якщо з кожним новим галуженням оцінка цільової функції “не погіршується”, то отриманий на певному кроці цілочисловий (дискретний) розв’язок – максимальний чи мінімальний розв’язок відповідної початкової задачі, а ні, то можливе повернення до однієї з попередніх кінцевих підмножин, що має кращу оцінку, ніж отримана на цьому етапі [9]. Тоді процес поділу виконують для цієї підмножини.

Множина допустимих розв’язків підзадачі співпадає з об’єднанням допустимих розв’язків підзадач, отриманих із неї шляхом декомпозиції. Якщо виконана умова відсікання означає, що підзадача не має розв’язків, то виключення цих підзадач не приводить до втрати допустимих розв’язків. Таким чином, будь-який максимальний чи мінімальний розв’язок задачі є також максимальним чи мінімальним розв’язком деякої підзадачі, що не відкидається. Тому для всіх підзадач, що не відкидаються, обраховуються максимальні чи мінімальні розв’язки, виключення їх з розгляду не приводить до втрати всіх максимальних чи мінімальних розв’язків задачі [3].

Розробка алгоритму на основі методу гілок та меж.

Схема алгоритму, що використовує метод гілок та меж, із застосуванням до задачі про двоетапний гільйотинний розкрій [5].

Введемо позначення: L – поточний список підмножин множини X .

КРОК 0. Аналіз множини X .

Знайти оцінку $\gamma(X) = (\text{ціла частина від ділення загальної площі головної дошки до площі найвигіднішої заготовки})$ та помножити на вартість цієї

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

заготовки. Хай X^* – множина допустимих розв'язків, що отриманні при оцінюванні множини X .

Якщо $X^* = \emptyset$, то вважаємо, що $L = \{X\}$, $f^* = +\infty$.

Якщо $X^* \neq \emptyset$ и для найкращого із цих розв'язків x^* виконується рівність $f(x^*) = \gamma(X)$, тоді x^* – розв'язок задачі, КІНЕЦЬ.

Якщо рівність $f(x^*) = \gamma(X)$ не виконується, тоді вважаємо, що $L = \{X\}$, $f^* = f(x^*)$.

КРОК 1. Аналіз списку.

Якщо список L не порожній, тоді перейти на КРОК 2, інакше – зупинити обрахунки. При цьому:

- якщо $f^* < +\infty$, тоді f^* – значення початкової задачі, а відповідний рекорду розв'язок – розв'язок задачі;
- якщо $f^* = +\infty$, тоді початкова задача не має допустимих розв'язків.

КРОК 2. Вибір множини для галуження.

Обрати зі списку L множину X^k за правилом:

$$X^k: \gamma(X^k) = \max \{ \gamma(X^j) \mid X^j \in L \}.$$

КРОК 3. Галуження.

Отримати множину $\beta(X^k) = \{X^{kj}\}$, $j = 1, 2, \dots, |\beta(X^k)|$. Де X^{kj} – це той розкрій, який був зроблений, тобто X^k , та до нього додаються інші види заготовок, тобто $j = 1$ – перша заготовка, $j = 2$ – друга заготовка і тому подібне (можна відрізати лише смугу, довжина якої не більше довжини залишку головної дошки).

КРОК 4. Аналіз нащадків.

Обрахувати оцінку для кожного елемента з $\beta(X^k) = \{X^{kj}\}$.

$\gamma(X^k)$ = сума прибутку від відрізаної смуги (отримана жадібним алгоритмом, що застосовується лише для розрізання по ширині) додати оцінку можливого прибутку від залишку головної дошки ((ціла частина від ділення площі головної дошки, що залишилася, до площі найвигіднішої заготовки) та помножити на прибутковість цієї заготовки).

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

КРОК 5. Коректування рекорду.

При оцінці множин X^{kj} допустимі розв'язки початкової задачі могли стати відомими. Нехай X^{kj*} – така множина допустимих розв'язків.

Якщо $X^* = \bigcup X^{kj*} \neq \emptyset$, тоді $f^* = \max\{f^*, \max_{x/x \in U X} \{f(x)\}\}$.

КРОК 6. Тест.

Сформувати новий список за правилом:

$$L = \{X^j | X^j \in (L \setminus X^k \cup \beta(X^k)), \gamma(X^j) < f^*\}.$$

При цьому з урахуванням того, що рекорд міг змінитися, зі списку будуть вилючені всі підмножини X^j , оцінка яких гірше, ніж у рекорду. Перехід на КРОК 1.

Приклад розв'язання задачі жадібним алгоритмом та алгоритмом на основі методу гілок та меж.

Розв'язання задачі жадібним алгоритмом зобразимо на рисунку 3.1.

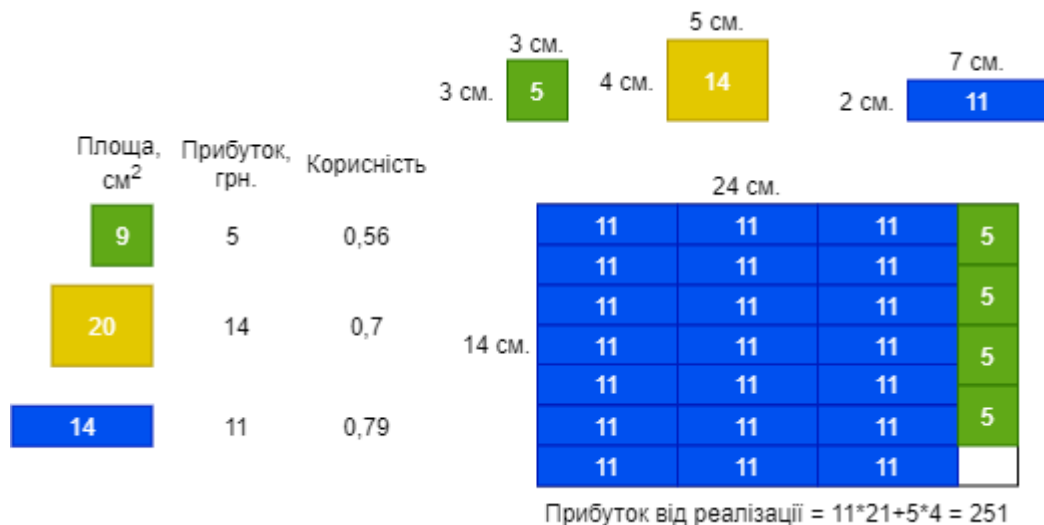


Рисунок 3.1 – Приклад розв'язання задачі жадібним алгоритмом
Знайдений сумарний прибуток – 251 гривня.

Розв'язання задачі алгоритмом на основі методу гілок та меж, коли дійшли до рекорду, зобразимо на рисунках 3.2 – 3.5.

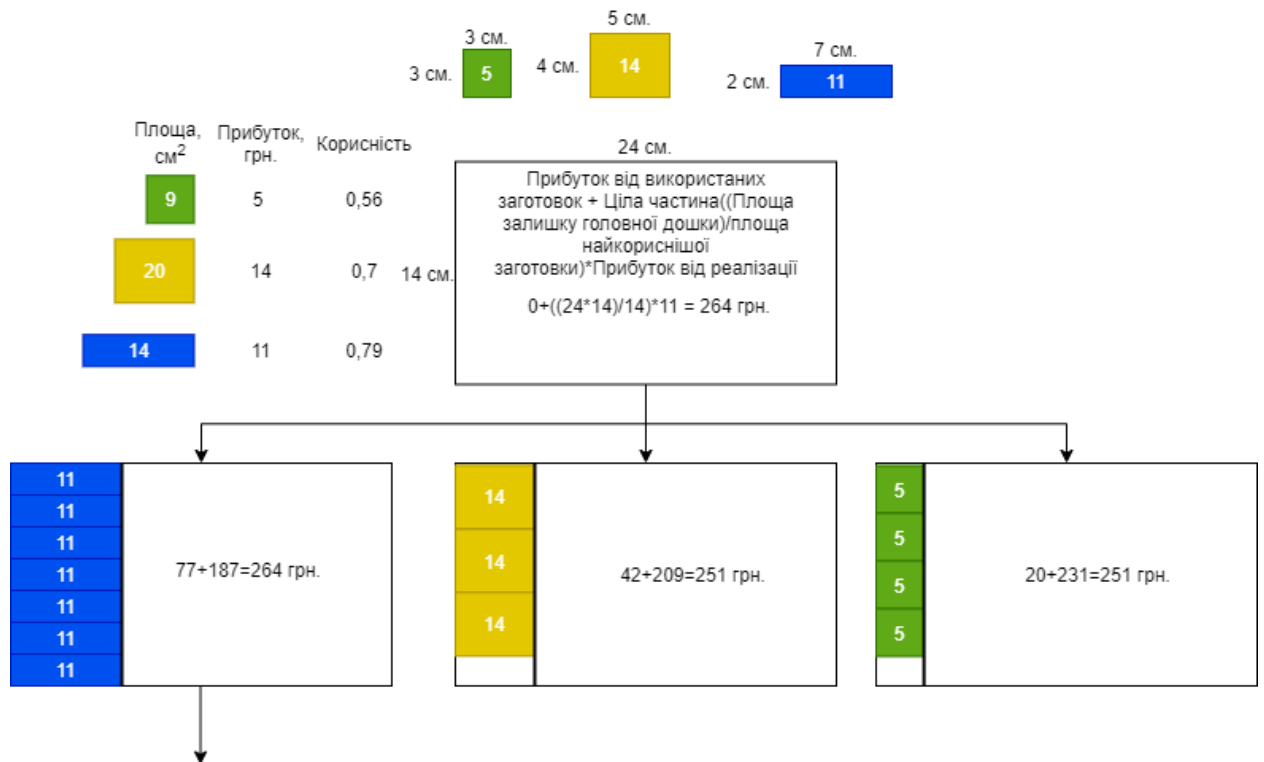


Рисунок 3.2 – Початок галуження для алгоритму на основі методу гілок та меж

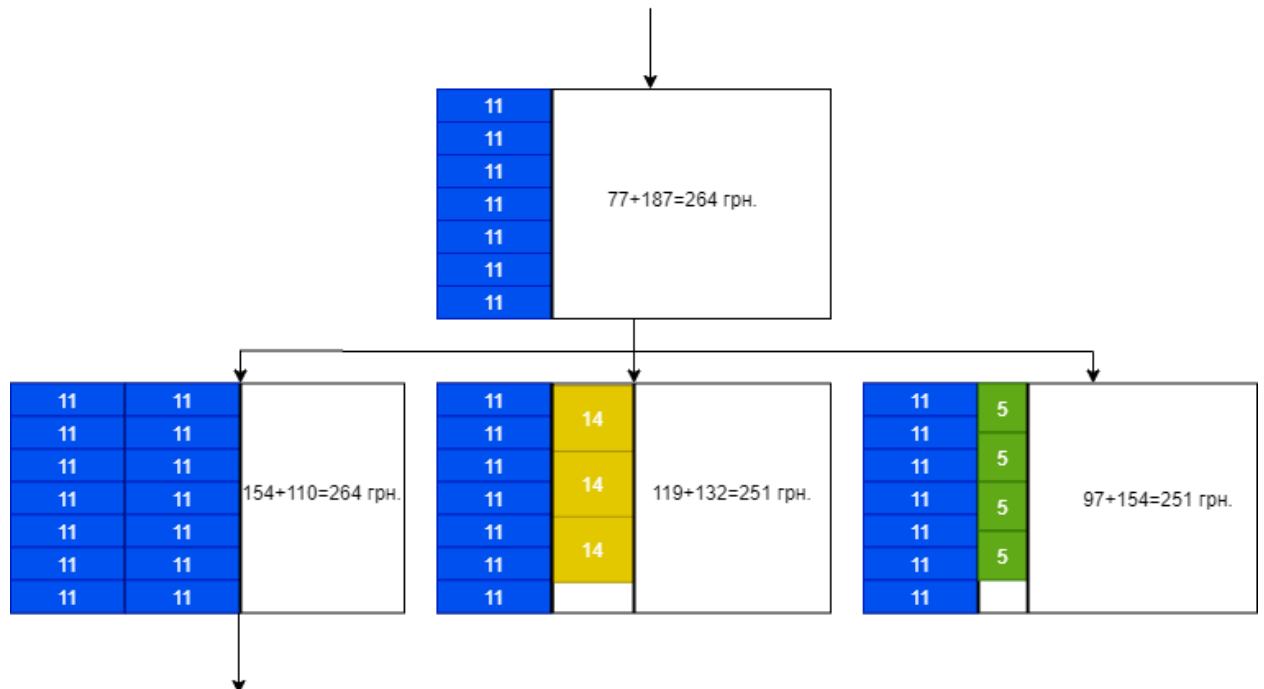


Рисунок 3.3 – Продовження галуження для алгоритму на основі методу гілок та меж

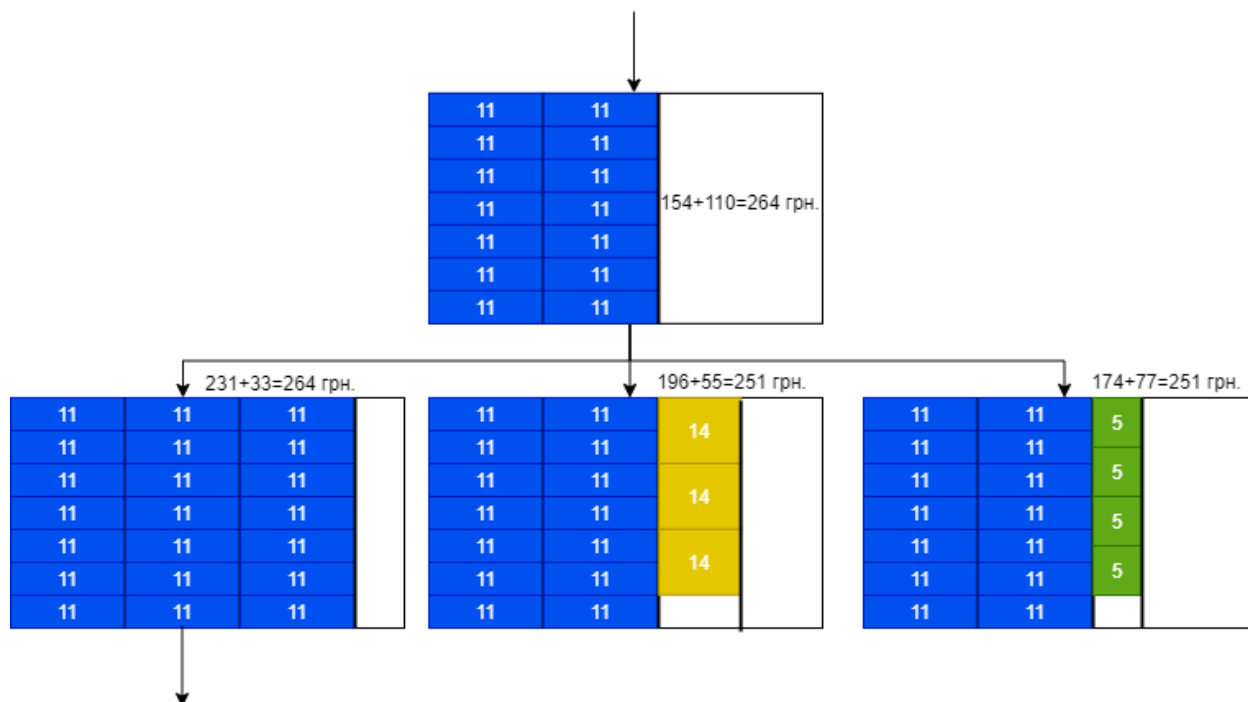


Рисунок 3.4 – Продовження галуження(2) для алгоритму на основі методу гілок та меж

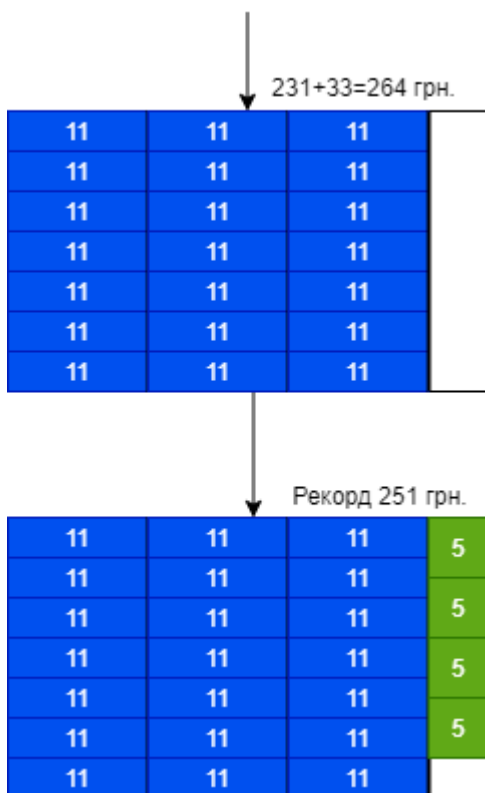


Рисунок 3.5 – Знайдено рекорд при галуженні для алгоритму на основі методу гілок та меж

Зобразимо процес галуження за допомогою графа на рисунку 3.6.

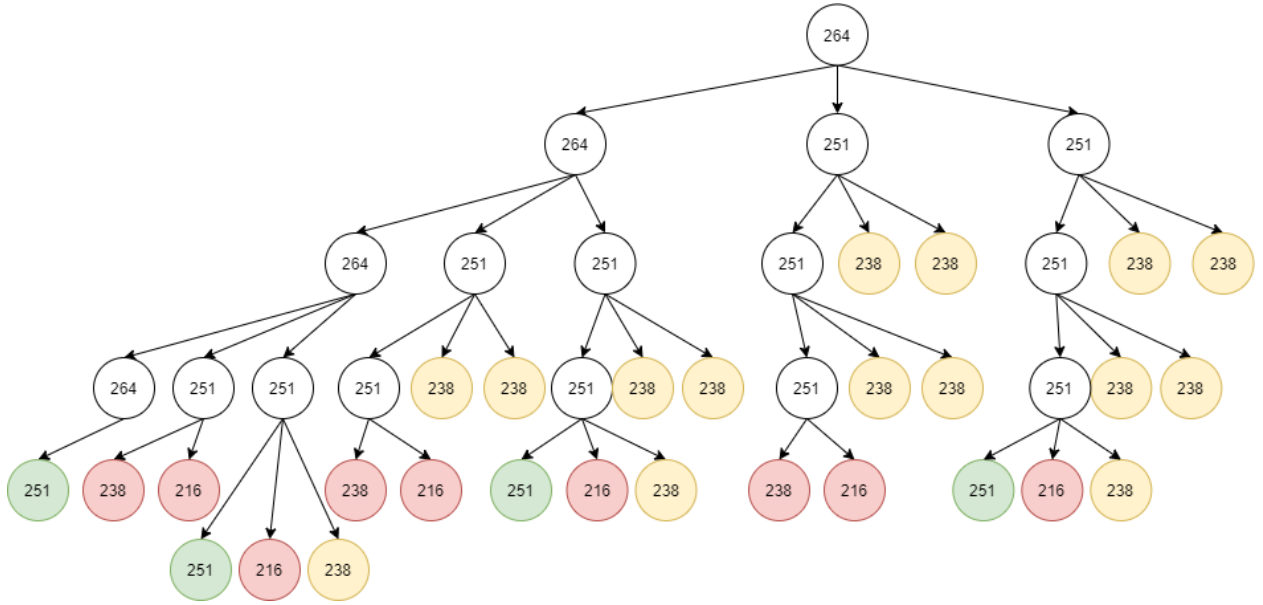


Рисунок 3.6 – Граф галуження

Зеленим кольором (значення 251, де кінцеві гілки) зображено рекорд, червоним (значення 216 та декілька 238, де кінцеві гілки) – кінцевий розкрій, гірший від рекорду, жовтим (значення 238) – незавершений розкрій, оцінка якого гірша від рекорду.

Альтернативний розкрій, зображений зеленим кольором, зобразимо на рисунку 3.7.

Рекорд 251 грн.				Альтернативний рекорд 251 грн.				Альтернативний рекорд 251 грн.				Альтернативний рекорд 251 грн.			
11	11	11	5	11	11	5	11	11	5	11	11	5	11	11	11
11	11	11	5	11	11	5	11	11	5	11	11	5	11	11	11
11	11	11		11	11		5	11		11	5		11	11	11
11	11	11	5	11	11	5	11	11	5	11	11	5	11	11	11
11	11	11	5	11	11	5	11	11	5	11	11	5	11	11	11
11	11	11		11	11		5	11		11	5		11	11	11
11	11	11		11	11		11	11		11	11		11	11	11

Рисунок 3.7 – Альтернативний розкрій

Отже, знайдено оптимальний план розкрою та його прибуток дорівнює 251.

Висновок до розділу

У розділі «Математичне забезпечення» описано змістовну постановку задачі, математичну постановку задачі розкрою, яка є базовою задачею для діяльності розкрійного цеху меблевого виробництва, обґрунтовано та описано методи розв'язання поставленої задачі.

Вказано складність задачі (NP-складна) та обґрунтовано методи її розв'язання. Через це обрано було два алгоритми для вирішення задачі: жадібний алгоритм, який легкий у реалізації, дає допустимий розв'язок, це швидкодія та може застосовуватись до задач великих розмірностей та алгоритм на основі методу гілок та меж, який гарно підходить для задач, коли кількість заготовок менша за п'ять, або відношення довжини головної дошки до довжин заготовок не більша за 100, дає точний розв'язок. Наведено алгоритмічне забезпечення задачі: ідеї алгоритмів, їх схеми, а також приклади вирішення задачі.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Програмний продукт написано завдяки мові програмування C # у середовищі розробки Microsoft Visual Studio використовуючи Windows Forms для графічного інтерфейсу та Microsoft SQL Server для роботи з базою даних. Порівняємо використані засоби розробки з аналогами.

По-перше, порівняємо середовище розробки Microsoft Visual Studio та Eclipse.

Microsoft Visual Studio – інтегроване середовище розробки, розроблено компанією Microsoft, яке дає можливість розробляти консольні програми та програми з графічним інтерфейсом, зокрема використовуючи технологію Windows Forms [11].

Переваги:

- продукт Microsoft, що гарантує зручне написання програми на мові програмування C # (розроблено Microsoft);
- наявна технологія Windows Forms;
- багато плагінів, які дають можливість функціонально розширити застосунок;
- добре підходить для створення програмного продукту для Windows.

Недоліки:

- для повного використання функціоналу необхідно придбати продукт;
- занадто багато функцій, які новий користувач не знайде через те, що необхідно знати повністю меню середовища [12].

Eclipse – інтегроване середовище розробки, розроблено Eclipse Foundation, з відкритим вихідним кодом [13]. Eclipse широко використовується для написання програм на мові програмування Java, але C # також підтримується середовищем.

Переваги:

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

- безкоштовне середовище розробки;
- багато плагінів, які користувачі самостійно створюють для середовища;
- компілятор, який працює швидше за компілятори конкурентів;
- можливість змінювати середовище, прибираючи зайві функції.

Недоліки:

- широка функціональність, яка для нового користувача буде зайвою;
- немає надійності плагінів, бо вони створюються суспільством, а не самими розробниками середовища [12].

По-друге, порівняємо мову програмування C # та Java.

C # – об'єктно-орієнтована мова програмування, розроблена компанією Microsoft в якості альтернативи мови програмування Java, яка подібна C-синтаксису [14].

Переваги:

- продукт Microsoft, що гарантує зручну роботу в Microsoft Visual Studio;
- об'єктно-орієнтований підхід;
- знаючи мови програмування C, C++ можна легко вивчити C #;
- велика кількість бібліотек [15].

Недоліки:

- погано підходить для написання програм не для Windows;
- не підтримується множинне успадкування класів [16].

Java – об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems, яка широко використовується в створенні мобільних застосунків та звичайних програм [17].

Переваги:

- об'єктно-орієнтований підхід;
- написаний код можна запустити будь-де, де наявна віртуальна машина Java;

- велика кількість бібліотек, написана звичайними людьми;
- добре підходить для використання многопоточності [18].

Недоліки:

- низька продуктивність;
- не підтримується множинне успадкування класів.

По-третє, порівняємо систему керування баз даними Microsoft SQL Server та MongoDB.

Microsoft SQL Server – система керування реляційними базами даних, розроблена компанією Microsoft, мова запитів Transact-SQL [19].

Переваги:

- продукт Microsoft, що гарантує зручну роботу з іншими продуктами Microsoft;
- наявні внутрішні та зовнішні ключі;
- керування даними завдяки SQL-запитів.

Недоліки:

- важко змінювати дані через зв'язність таблиць;
- може зайняти всі доступні ресурси [20].

MongoDB – документ-орієнтована система керування баз даними, розроблена компанією MongoDB, не використовує SQL-запити [21].

Переваги:

- дозволяє легко змінювати структуру бази даних;
- швидко працює з великими обсягами даних;
- підтримка JSON [22].

Недоліки:

- дані майже не зв'язні;
- не використовує SQL-запити.

Отже, порівнявши використані засоби розробки з аналогами дійсно можна переконатися у правильності вибору, бо всі обрані засоби розроблені компанією Microsoft, що дозволяє зручно розробляти програмний продукт.

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Для вірного функціонування програмного продукту вимогами до технічного забезпечення є комп'ютер та доступ до бази даних.

4.3 Архітектура програмного забезпечення

Для того щоб описати архітектуру програмного забезпечення, необхідно навести діаграму класів, послідовності, компонентів та дати їх опис.

4.3.1 Діаграма класів

Діаграма класів необхідна, щоб зобразити внутрішню структуру системи, тобто відобразити класи системи, їх методи та відношення між класами. Класи в майбутньому будуть об'єктами. Методи класу – це поведінка об'єкта. Відношення між класами – це відношення між об'єктами. Схема структурна класів наведена в графічному матеріалі.

Опис класів:

- «Program» – точка входу в застосунок;
- «MainForm» – створює вікно для обрання методу роботи з програмою;
- «BDWork» – створює вікно для роботи з базою даних;
- «UsualWork» – створює вікно для роботи в діалоговому режимі;
- «BarsDB» – створює об'єкт для занесення даних заготовок замовника з бази даних;
- «HelpBD» – створює вікно з інструкцією користування для вікна роботи з базою даних;
- «DataCustomer» – створює вікно для заповнення даних замовника;
- «HelpUsual» – створює вікно з інструкцією користування для вікна роботи в діалоговому режимі;
- «Author» – створює вікно з інформацією про автора;

- «GreedyAlgorithm» – створює об'єкт для роботи жадібного алгоритму;
- «MGtaM» – створює об'єкт для роботи алгоритму на основі методу гілок та меж;
- «Node» – створює об'єкт (вершину) для галуження при роботі алгоритму на основі методу гілок та меж;
- «Workpiece» – створює об'єкт (заготовку) для подальшої роботи алгоритмів.

4.3.2 Діаграма послідовності

Діаграма послідовності описує логіку сценарію використання завдяки об'єктам та повідомленням. Об'єкти відображають взаємодію в рамках сценарію шляхом обміну повідомленнями та зворотними повідомленнями, які містять результат. Схема структурна послідовності наведена в графічному матеріалі.

В схемі структурній послідовності маємо два варіанти використання: варіант розкрою на основі даних замовників та варіант розкрою в діалоговому режимі. Розглянемо на прикладі, коли обрано послідовність для варіанту розкрою на основі даних замовників: менеджер хоче змінити дані замовника.

Об'єкт «Менеджер» відкриває вікно для роботи з даними замовників (об'єкт «Дані замовників»). В таблицях об'єкт «Менеджер» змінює дані об'єкта «Дані замовників» та зберігає в базі даних (об'єкт «База даних»). Потім з об'єкта «База даних» завантажуються нові дані в таблиці об'єкта «Дані замовників». Об'єкт «Менеджер» бачить зміни в таблицях об'єкта «Дані замовників».

Далі розглянемо на прикладі, коли обрано послідовність для варіанту розкрою в діалоговому режимі: менеджер хоче переглянути звіт замовника.

Об'єкт «Менеджер» відкриває вікно для роботи в діалоговому режимі (об'єкт «Діалоговий режим»). Об'єкт «Менеджер» обирає звіт якого замовника переглянути (об'єкт «Звіт замовника»). Об'єкт «Звіт замовника»

відображає звіт замовника завдяки об'єкту «Діалоговий режим». Об'єкт «Менеджер» бачить звіт завдяки об'єкту «Діалоговий режим».

4.3.3 Діаграма компонентів

Діаграма компонентів зображує програмну систему, яка поділена на структурні компоненти та зв'язки між компонентами. Схема структурна компонентів наведена в графічному матеріалі.

Схема має сім компонентів, але деякі компоненти об'єднані в один компонент: «Work option», «Algorithms» та «Database».

- «Work option» – головний компонент, який керує програмним продуктом;
- «Algorithms» – компонент, який дає можливість робити розрахунки обраним алгоритмом;
- «Database» – компонент для роботи програмного застосунку з базою даних.

4.3.4 Специфікація функцій

Для того, щоб розуміти програмний код необхідно навести специфікацію функцій. Назва методів та їх опис наведено в таблиці 4.1.

Таблиця 4.1 – Специфікація функцій

Назва	Опис
DataLoad()	Завантажує дані в таблиці з бази даних
ClearFunction()	Очищує поля графічного інтерфейсу
VisOrEnabFunction(bool bl)	Скриває чи блокує елементи графічного інтерфейсу
UtilityCalc(List<Workpiece> workpieces)	Підраховує корисність заготовок
CalculateResults(CancellationToken token, string algo)	Розраховує прибуток та план розкрою

Продовження таблиці 4.1.

Назва	Опис
CheckRes()	Перевіряє наявність результату розрахунків в таблицях для замовника
GetBars(List<Workpiece> lst, string res)	Розташовує кількість використаних заготовок в тому порядку, в якому заготовки були введені для розрахунків та записує результат у відповідне поле
Work(int widthB, int length, List<Workpiece> workpieces)	Розрахунок прибутку та кількості використаних заготовок жадібним алгоритмом
CreateNode(int lenght, int mark, int addedvalue)	Створення вершини для методу гілок та меж
Max(int a, int b)	Повертає максимальне значення
GreedyComponent(int length_left, int width)	Підрахунок отриманого прибутку від розкрою вертикальної смуги жадібним алгоритмом для алгоритму на основі методу гілок та меж
AddingValues(Node oldnode, Node newnode)	Додання кількості використаних заготовок для вершини
AlgorithmWorking(List<Workpiece> workpieces, int width, Node node)	Розрахунок прибутку та кількості використаних заготовок алгоритмом на основі методу гілок та меж

4.4 Опис звітів

Менеджер завдяки програмного продукту формує два види звітів: звіт замовника та завдання на розкрій. Приклад звіту замовника «Margo Cool MGtaM» наведено на рисунку 4.4.

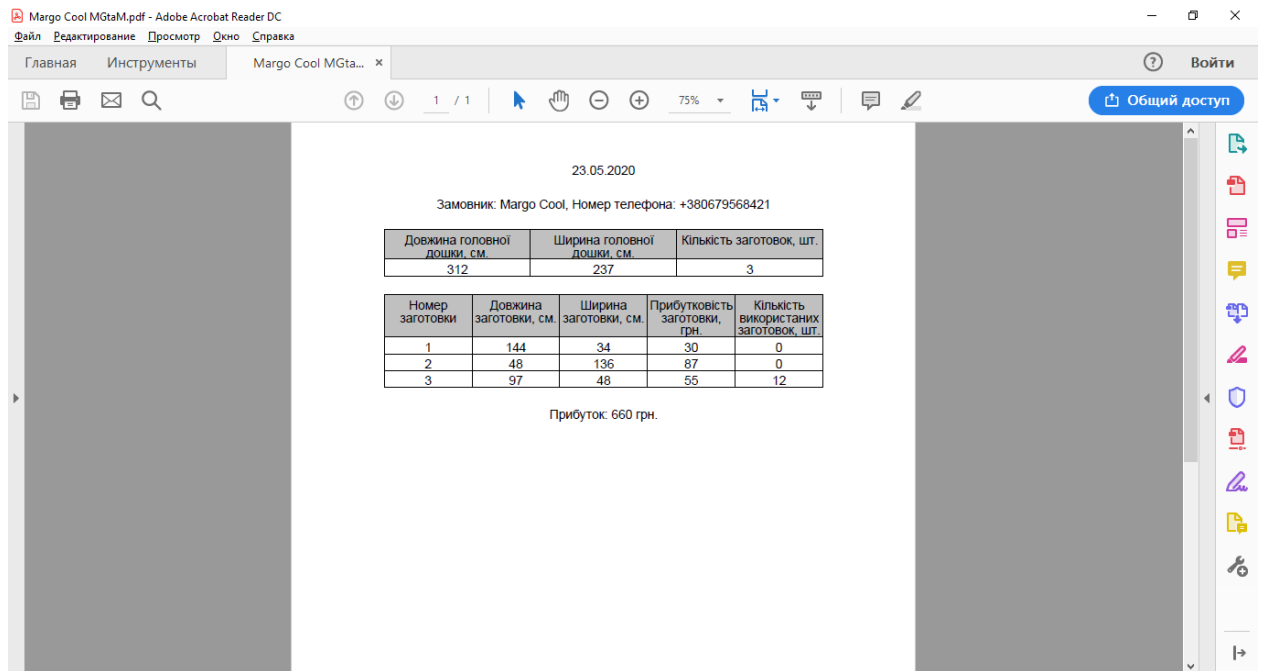


Рисунок 4.4 – Звіт для замовника Margo Cool

В цьому звіті наведені всі вхідні дані та вихідні, тобто результати розкрою, а також інформація про замовника та дата розрахунків. Приклад завдання на розкрій «Margo Cool MGtaM» наведено на рисунку 4.5.

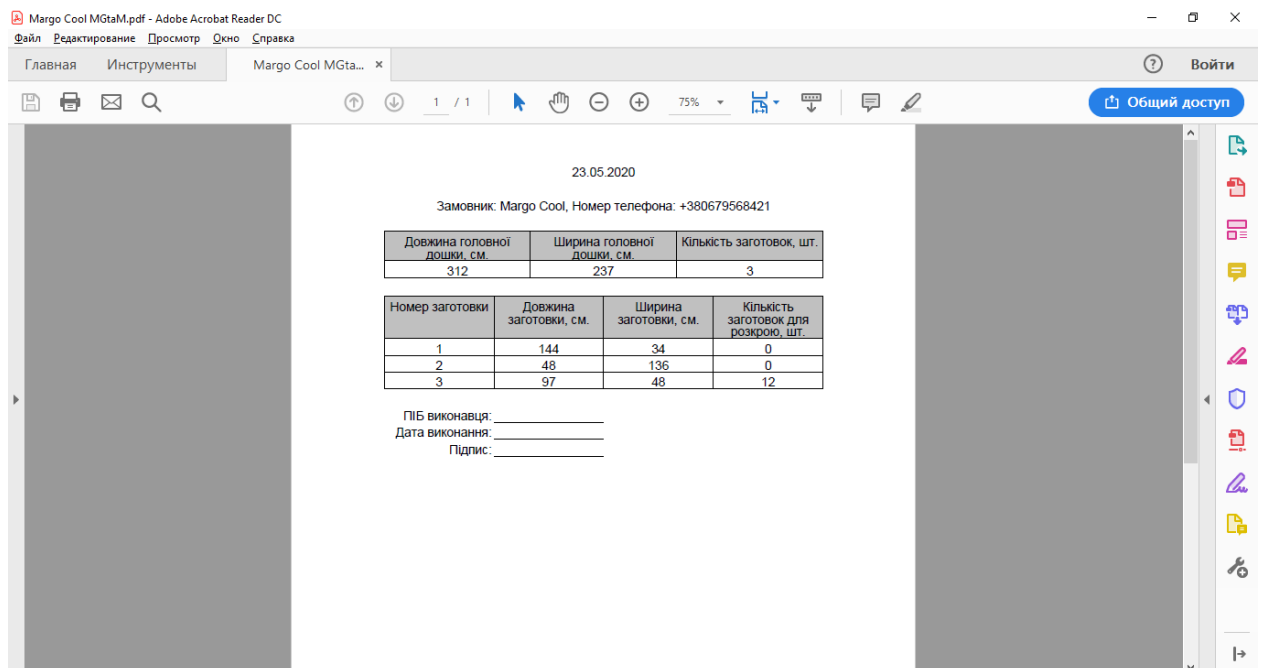


Рисунок 4.5 – Завдання на розкрій

Так само тут знаходиться дата розрахунку, дані замовника, вхідні дані для розкрою та кількість заготовок необхідна для розкрою. А також поля, які

необхідно заповнити розкрійнику для того, щоб було розуміння, що розкрій виконано та людина виконала свою роботу.

Висновок до розділу

У розділі «Програмне та технічне забезпечення» описано засоби розробки, вимоги до технічного забезпечення, архітектуру програмного забезпечення та звіти.

Було порівняно засоби розробки, які були обрані для написання програмного продукту та аналоги, які могли були використані. Обрані засоби розробки створені компанією Microsoft, тому гарно поєднуються.

Для вірного функціонування програмного продукту вимогами до технічного забезпечення є комп'ютер та доступ до бази даних. Були наведені та описані діаграми класів, послідовності та компонентів для розуміння архітектури програмного забезпечення.

Описані та наведені звіти, які формує менеджер завдяки програмі.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Для початку роботи з програмним продуктом потрібно запустити файл DiplomaProgram.exe. Далі відкривається головне вікно програми, де можна обрати один з варіантів використання: «Вибір варіанту розкрою на основі даних замовників(з бази даних)» та «Вибір варіанту розкрою в діалоговому режимі». Також є кнопка «Вихід» для виходу з програми. Головне вікно зображено на рисунку 5.1.

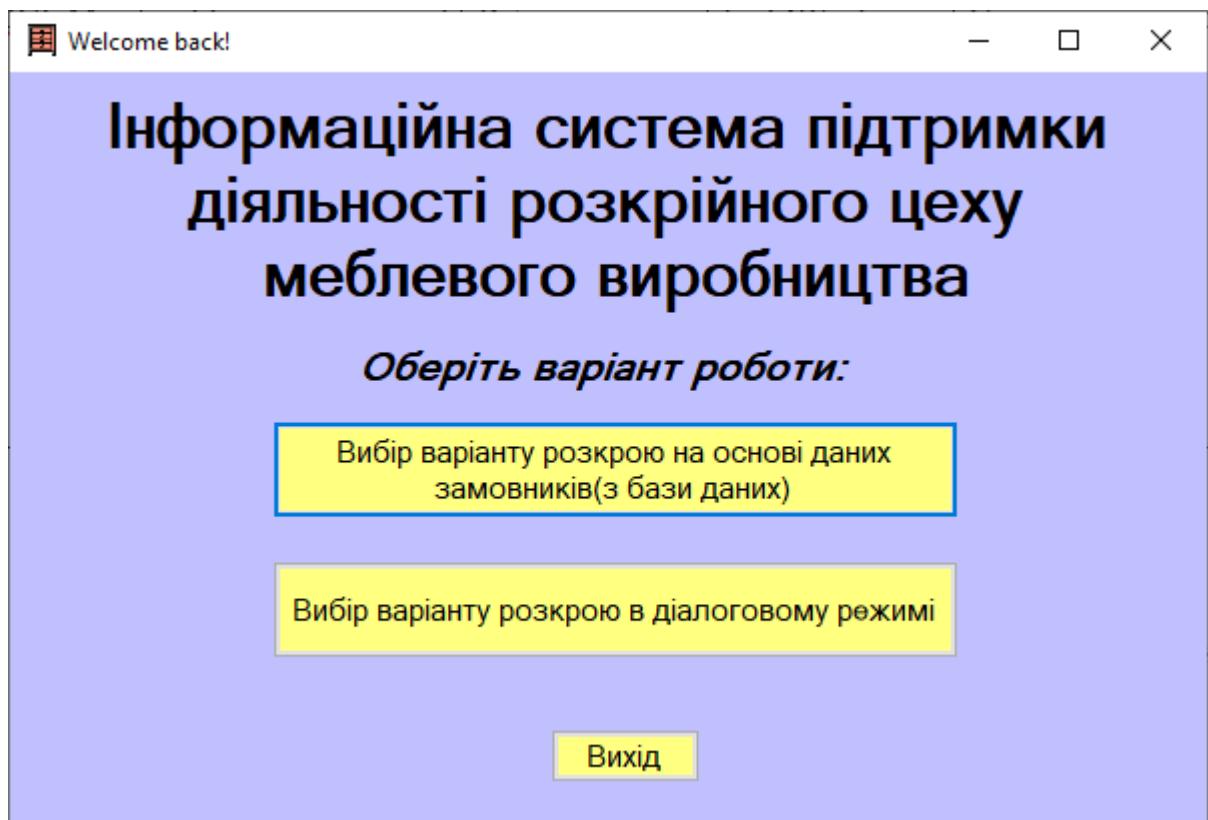


Рисунок 5.1 – Головне вікно програми

Далі натискаємо «Вибір варіанту розкрою на основі даних замовників(з бази даних)», якщо бажаємо попрацювати з даними замовників та розрахувати для них прибуток або змінити дані замовників. Відповідне вікно зображено на рисунку 5.2.

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
1	Oleh	Mospak	671112223	24	14	3	
2	Vasily	Vaskov	671564865	20	30	3	
3	Dmitriy	Soniv	638789856	10000	200	2	
4	Danil	Sizov	695648532	230	30	3	
5	Kiril	Nice	637895425	160	45	3	
6	Margo	Cool	679568421	312	237	3	

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
1	1	1	3	3	5	
2	1	2	4	5	14	
3	1	3	2	7	11	
4	2	1	5	10	10	
5	2	2	10	10	2	
6	2	3	2	5	8	

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
15	6	Greedy	522	
16	6	MGtaM	660	
17	16	MGtaM	527	

Робота з таблицями

Введіть Id замовника для формування плану:

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
42	6	Greedy	1	0	
43	6	Greedy	2	6	
44	6	Greedy	3	0	
45	6	MGtaM	1	0	
46	6	MGtaM	2	0	
47	6	MGtaM	3	12	

Обрати звіт для перегляду:

Оберіть алгоритм для розрахунку:

Прибуток: Кількість використаних заготовок:

Робота в діалоговому режимі

Рисунок 5.2 – Вікно для роботи з даними замовників

Якщо не знаєте що потрібно роботи в вікні та в якому порядку це роботи, необхідно натиснути на пункт меню «Допомога» та з'явиться вікно з інструкцією користувача. Вікно «Допомога» зображено на рисунку 5.3.

Help

Інструкція користувача

1. Якщо бажаєте попрацювати з таблицями натисніть кнопку "Робота з таблицями".
 - 1.1. З'явилися кнопки "Додати", "Зберегти", "Видалити".
 - 1.2. Щоб додати замовника чи заготовки замовника натисніть кнопку "Додати"(заповніть всі дані, окрім поля Id) та натисніть кнопку "Зберегти".
 - 1.3. Щоб змінити дані в таблицях натисніть на відповідне поле два рази, змініть дані та натисніть кнопку "Зберегти".
 - 1.4. Щоб видалити дані з таблиць оберіть рядок в таблиці та натисніть кнопку "Видалити".
 - 1.5. Якщо не вдається щось зробити, то з'явиться відповідне повідомлення про помилку.
2. Введіть Id замовника у відповідне поле для формування плану розкрою.
3. Натисніть кнопку "OK".
 - 3.1. Якщо всі дані коректні, то з'явиться фрагмент з вхідними даними.
 - 3.2. Якщо дані не коректні, то з'явиться відповідне повідомленням про помилку.
4. Оберіть алгоритм у відповідному полі для підрахунку прибутку та кількості використаних заготовок.
 - 4.1. Жадібний алгоритм дає допустимий результат, але не обов'язково оптимальний.
 - 4.2. Метод гілок та меж(МГ та М) дає оптимальний результат, але, наприклад, при довжині головної дошки 2 м та заготовок 2 см знадобиться хвилин 10 для підрахунку.
5. Натисніть кнопку "Порахувати" та дочекайтесь результату.
 - 5.1. Якщо виникла помилка, то з'явиться відповідне повідомлення.
- 5.2. Якщо бажаєте зупинити підрахунок натисніть кнопку "Зупинити".
6. Передивіться результати у відповідних полях та в таблицях.
 - 6.1. Якщо бажаєте сформувати звіт для замовника натисніть кнопку "Сформувати звіт".
 - 6.2. Якщо бажаєте сформувати завдання на розкрій натисніть кнопку "Сформувати завдання на розкрій".
7. Якщо бажаєте передивитися звіт, оберіть звіт у відповідному полі та натисніть кнопку "Відкрити звіт".
8. Якщо бажаєте спробувати альтернативні варіанти для замовника натисніть кнопку "Робота в діалоговому режимі".

Рисунок 5.3 – Вікно з інструкцією користувача для вікна «Вибір варіанту розкрою на основі даних замовників(з бази даних)»

Перечитавши інструкцію закриваємо це вікно.

Також можна подивитися інформацію про автора, для цього необхідно натиснути на пункт меню «Про автора» на відкриється вікно, зображене на рисунку 5.4.

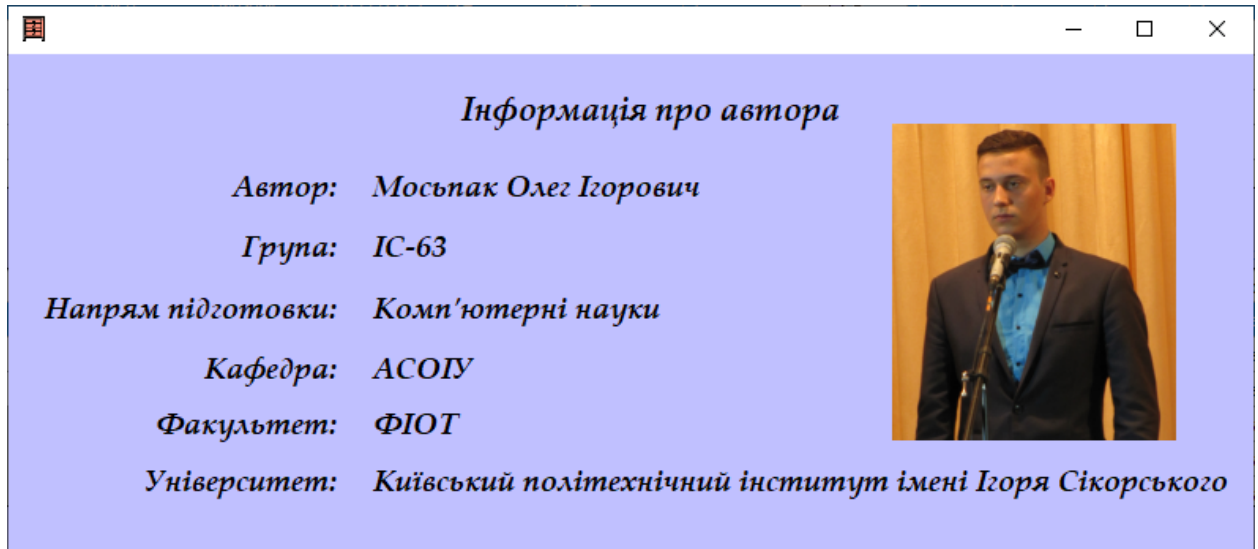


Рисунок 5.4 – Вікно з інформацією про автора

Подивившись інформацію зачиняємо вікно.

На сторінці «Вибір варіанту розкрою на основі даних замовників(з бази даних)» представлені чотири таблиці, на яких знаходиться інформація про замовників, заготовок, прибуток та кількість використаних заготовок. Якщо необхідно змінити інформацію про замовника чи параметри заготовок, натисніть на кнопку «Робота з таблицями». Змінене вікно зображено на рисунку 5.5.

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
▶	1	Oleh	Mospak	671112223	24	14	3
	2	Vasiliy	Vaskov	671564865	20	30	3
	3	Dmitriy	Soniv	638789856	10000	200	2
	4	Danil	Sizov	695648532	230	30	3
	5	Kiril	Nice	637895425	160	45	3
	6	Margo	Cool	679568421	312	237	3

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	1	1	1	3	3	5
	2	1	2	4	5	14
	3	1	3	2	7	11
	4	2	1	5	10	10
	5	2	2	10	10	2
	6	2	3	2	5	8

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
▶	15	6	Greedy	522
	16	6	MGtaM	660
	17	16	MGtaM	527

Видалити

Введіть Id замовника для формування плану: ОК

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
▶	42	6	Greedy	1	0
	43	6	Greedy	2	6
	44	6	Greedy	3	0
	45	6	MGtaM	1	0
	46	6	MGtaM	2	0
	47	6	MGtaM	3	12

Видалити

Оберіть алгоритм для розкрою:

Поранувати Сформувати звіт Сформувати завдання на розкрій

Зупинити Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Прибуток: Кількість використаних заготовок:

Рисунок 5.5 – Вікно після натискання кнопки «Робота з таблицями»

Робота з таблицею замовників:

- редагування даних замовника: змінимо дані для замовника з останнього поля. Щоб, наприклад, змінити довжину головної дошки в полі «Length_Main» потрібно двічі натиснути на значення в цій колонці. Змінюємо значення на необхідне та натискаємо «Enter». Після цього натискаємо кнопку «Зберегти» для збереження інформації в таблиці та занесення змін до бази даних. Результат зображено на рисунку 5.6;

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
	6	Margo	Cool	679568421	312	237	3
	9	Nastya	Bus	632322152	380	432	3
	10	Fedor	Car	935558825	362	174	2
	12	Nikita	Last	685462352	300	200	2
▶	16	Danilo	U	685642535	230	237	3

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	1	1	1	3	3	5
	2	1	2	4	5	14
	3	1	3	2	7	11
	4	2	1	5	10	10
	5	2	2	10	10	2
	6	2	3	2	5	8

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
▶	15	6	Greedy	522
	16	6	MGtaM	660
	17	16	MGtaM	527

Видалити

Введіть Id замовника для формування плану: OK

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
▶	42	6	Greedy	1	0
	43	6	Greedy	2	6
	44	6	Greedy	3	0
	45	6	MGtaM	1	0
	46	6	MGtaM	2	0
	47	6	MGtaM	3	12

Видалити

Оберіть алгоритм для розрахунку:

Прибуток: Кількість використаних заготовок:

Поранувати Сформувати звіт Сформувати завдання на розкрій

Зупинити Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Рисунок 5.6 – Змінене та збережене значення довжини головної дошки для замовника з Id рівним 16

- додавання даних замовника: натискаємо на кнопку «Додати» під таблицею «Замовники». Додається рядок, де необхідно ввести дані (поле Id не заповнюємо). Вводимо дані та натискаємо кнопку «Зберегти». Результат зображено на рисунку 5.7;

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
	9	Nastya	Bus	632322152	380	432	3
	10	Fedor	Car	935558825	362	174	2
	12	Nikita	Last	685462352	300	200	2
	16	Danilo	U	685642535	230	237	3
▶	17	Nastya	Second	658528825	325	207	2

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	1	1	1	3	3	5
	2	1	2	4	5	14
	3	1	3	2	7	11
	4	2	1	5	10	10
	5	2	2	10	10	2
	6	2	3	2	5	8

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
▶	15	6	Greedy	522
	16	6	MGtaM	660
	17	16	MGtaM	527

Видалити

Введіть Id замовника для формування плану: OK

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
▶	42	6	Greedy	1	0
	43	6	Greedy	2	6
	44	6	Greedy	3	0
	45	6	MGtaM	1	0
	46	6	MGtaM	2	0
	47	6	MGtaM	3	12

Видалити

Оберіть алгоритм для розрахунку:

Прибуток: Кількість використаних заготовок:

Поранувати Сформувати звіт Сформувати завдання на розкрій

Зупинити Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Рисунок 5.7 – Додана інформація про нового замовника

- видалення даних замовника: обираємо рядок, один раз натискаємо на нього та потім натискаємо на кнопку «Видалити». Далі натискаємо кнопку «Зберегти», результат зображено на рисунку 5.8.

Work with data of customers

Допомога Про автора

Вибір варіанту розкрою на основі даних замовників

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
	6	Margo	Cool	679568421	312	237	3
	9	Nastya	Bus	632322152	380	432	3
	10	Fedor	Car	93558825	362	174	2
	12	Nikita	Last	685462352	300	200	2
	16	Danilo	U	685642535	230	237	3

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
	1	1	1	3	3	5
	2	1	2	4	5	14
	3	1	3	2	7	11
	4	2	1	5	10	10
	5	2	2	10	10	2
	6	2	3	2	5	8

Додати Зберегти Видалити Робота з таблицями

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
	15	6	Greedy	522
	16	6	MGtaM	660
	17	16	MGtaM	527

Введіть Id замовника для формування плану: OK

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
	42	6	Greedy	1	0
	43	6	Greedy	2	6
	44	6	Greedy	3	0
	45	6	MGtaM	1	0
	46	6	MGtaM	2	0
	47	6	MGtaM	3	12

Додати Зберегти Видалити

Обрати алгоритм для розрахунку:

Прибуток: Кількість використаних заготовок:

Видалити

Обрати звіт для перегляду: Відкрити звіт

Порахувати Зупинити Сформувати звіт Сформувати завдання на розкрій

Робота в діалоговому режимі

Рисунок 5.8 – Результат видалення даних замовника

Отже, зображені всі можливості роботи з таблицями.

Для формування плану розкрою необхідно обрати замовника для якого бажаємо сформувати план розкрою, запам'ятовуємо значення Id. Далі у відповідне поле вводимо це значення та натискаємо кнопку «ОК». Змінене вікно зображено на рисунку 5.9.

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_1
1	Oleh	Mospak	671112223	24	14	3	
2	Vasily	Vaskov	671564865	20	30	3	
3	Dmitry	Soniv	638789856	10000	200	2	
4	Danil	Sizov	695648532	230	30	3	
5	Kiril	Nice	637895425	160	45	3	
6	Margo	Cool	679568421	312	237	3	

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
1	1	1	3	3	3	5
2	1	2	4	5	14	
3	1	3	2	7	11	
4	2	1	5	10	10	
5	2	2	10	10	2	
6	2	3	2	5	8	

Додати Зберегти Видалити

Таблиця "Результати доходу"

Id	Customer_Id	Algorithm	Total_Income

Введіть Id замовника для формування плану: 1 OK

Вхідні дані

Довжина головної дошки: 24

Роздільник - пробіл

Довжини заготовок: 3 4 2

Ширини головної дошки: 14

Ширини заготовок: 3 5 7

Кількість заготовок: 3

Прибутковість заготовок: 5 14 11

Вибрати алгоритм для розрахунку:

Прибуток: Кількість використаних заготовок:

Вибрати

Таблиця "Результати кількості заготовок"

Id	Customer	Algorithm	Bar	Count_Ba

Вибрати

Поррахувати Сформувати звіт Сформувати завдання на розкрій

Зупинити Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Рисунок 5.9 – Сформований план розкрою для обраного замовника

Як бачимо, поля які стосуються замовника стали виділеними та з'явився фрагмент з вхідними даними, які записані у блоці «Вхідні дані».

Вхідні дані заповнено відповідними значеннями з таблиць «Замовники» та «Заготовки». Дані, які стосуються параметрів заготовок – заповнені через пробіл. Перше число – дані першої заготовки, друге число – дані другої заготовки і так далі. Як бачимо, всі поля з вхідними даними і поля «Прибуток» та «Кількість використаних заготовок» є неактивні, бо туди інформація заноситься автоматично.

Далі обираємо алгоритм для розрахунку, є два алгоритми: «Жадібний алгоритм» та «Метод гілок та меж». Обираємо «Жадібний алгоритм» у полі та натискаємо на кнопку «Поррахувати». Результат зображено на рисунку 5.10.

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_1
▶	1	Oleh	Mospak	671112223	24	14	3
	2	Vasily	Vaskov	671564865	20	30	3
	3	Dmitry	Soniv	638789856	10000	200	2
	4	Danil	Sizov	695648532	230	30	3
	5	Kiril	Nice	637895425	160	45	3
	6	Margo	Cool	679568421	312	237	3

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	1	1	1	3	3	5
	2	1	2	4	5	14
	3	1	3	2	7	11
	4	2	1	5	10	10
	5	2	2	10	10	2
	6	2	3	2	5	8

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
▶	18	1	Greedy	264

Введіть Id замовника для формування плану: 1 OK

Вхідні дані: Довжина головної дошки: 24 Роздільник - пробіл: 3 4 2

Ширину головної дошки: 14 Ширину заготовок: 3 5 7

Кількість заготовок: 3 Прибутковість заготовок: 5 14 11

Оберіть алгоритм для розрахунку: Жадібний алгоритм

Прибуток: 264 Кількість використаних заготовок: 0 0 24

Таблиця "Результати кількості заготовок"

	Id	Customer_	Algorithm	Bar	Count_Ba
▶	51	1	Greedy	1	0
	52	1	Greedy	2	0
	53	1	Greedy	3	24

Видалити

Порахувати Зупинити Сформувати звіт Сформувати завдання на розкрій

Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Рисунок 5.10 – Результат роботи жадібного алгоритму для сформованого плану обраного замовника

У таблиці «Результати доходу» та «Результати кількості заготовок» занесені результати роботи жадібного алгоритму (в полях «Algorithm» значення «Greedy»). Також в цих таблицях виділені поля замовника для якого робились обчислення. Порахуємо прибуток завдяки алгоритму «Метод гілок та меж». Також обираємо його та натискаємо кнопку «Порахувати». Якщо не маємо бажання чекати завершення підрахунку, то натискаємо кнопку «Зупинити». Результат розрахунків зображено на рисунку 5.11.

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_1
▶	1	Oleh	Mospak	671112223	24	14	3
	2	Vasily	Vaskov	671564865	20	30	3
	3	Dmitry	Soniv	638789856	10000	200	2
	4	Danil	Sizov	695648532	230	30	3
	5	Kiril	Nice	637895425	160	45	3
	6	Margo	Cool	679568421	312	237	3

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	1	1	1	3	3	5
	2	1	2	4	5	14
	3	1	3	2	7	11
	4	2	1	5	10	10
	5	2	2	10	10	2
	6	2	3	2	5	8

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
▶	18	1	Greedy	264
	19	1	MGtaM	264

Введіть Id замовника для формування плану: 1 OK

Вхідні дані: Довжина головної дошки: 24 Роздільник - пробіл: 3 4 2

Ширину головної дошки: 14 Ширину заготовок: 3 5 7

Кількість заготовок: 3 Прибутковість заготовок: 5 14 11

Оберіть алгоритм для розрахунку: Метод гілок та меж

Прибуток: 264 Кількість використаних заготовок: 0 0 24

Видалити

Порахувати Сформувати звіт Сформувати завдання на розкрій

Зупинити Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
	52	1	Greedy	2	0
	53	1	Greedy	3	24
▶	54	1	MGtaM	1	0
	55	1	MGtaM	2	0
	56	1	MGtaM	3	24

Видалити

Рисунок 5.11 – Результат роботи алгоритму на основі методу гілок та меж для сформованого плану обраного замовника

Також з'явилися нові поля в таблицях «Результати доходу» та «Результати кількості заготовок», вони виділені та значення в полі «Algorithm» – «MGtaM». Як бачимо обидва алгоритми дали один і той самий результат. Порахуємо двома алгоритмами для замовника з Id – 6. Результат зображено на рисунку 5.12.

Work with data of customers

Допомога Про автора

Вибір варіанту розкрою на основі даних замовників

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_1
▶	6	Margo	Cool	679568421	312	237	3
	9	Nastya	Bus	632322152	380	432	3
	10	Fedor	Car	935558825	362	174	2
	12	Nikita	Last	685462352	300	200	2
	16	Danilo	U	685642535	230	237	3

Додати Зберегти Видалити Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	15	6	1	144	34	30
	16	6	2	48	136	87
	17	6	3	97	48	55
	18	9	1	86	89	46
	19	9	2	85	20	29
	20	9	3	25	115	51

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
	18	1	Greedy	264
	19	1	MGtaM	264
	20	6	Greedy	522
▶	21	6	MGtaM	660

Видалити

Введіть Id замовника для формування плану: 6 OK

Вхідні дані: Довжина головної дошки: 312 Роздільник - пробіл: 144 48 97

Ширину головної дошки: 237 Ширину заготовок: 34 136 48

Кількість заготовок: 3 Прибутковість заготовок: 30 87 55

Оберіть алгоритм для розкрою: Метод гілок та меж

Прибуток: 660 Кількість використаних заготовок: 10 12

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
	58	6	Greedy	2	6
	59	6	Greedy	3	0
▶	60	6	MGtaM	1	0
	61	6	MGtaM	2	0
	62	6	MGtaM	3	12

Видалити

Поранувати Зупинити Сформувати звіт Сформувати завдання на розкрій

Робота в діалоговому режимі

Обрати звіт для перегляду: Відкрити звіт

Рисунок 5.12 – Результат роботи алгоритмів для замовника з Id – 6

Бачимо, що алгоритм на основі методу гілок та меж дав результат – 660, а жадібний алгоритм – 522. Отже, метод гілок та меж дав кращий результат, бо це точний метод.

Якщо необхідно сформувати звіт для замовника, натискаємо кнопку «Сформувати звіт» та бачимо вікно для вибору назви файлу та шляху. Вікно збереження зображено на рисунку 5.13.

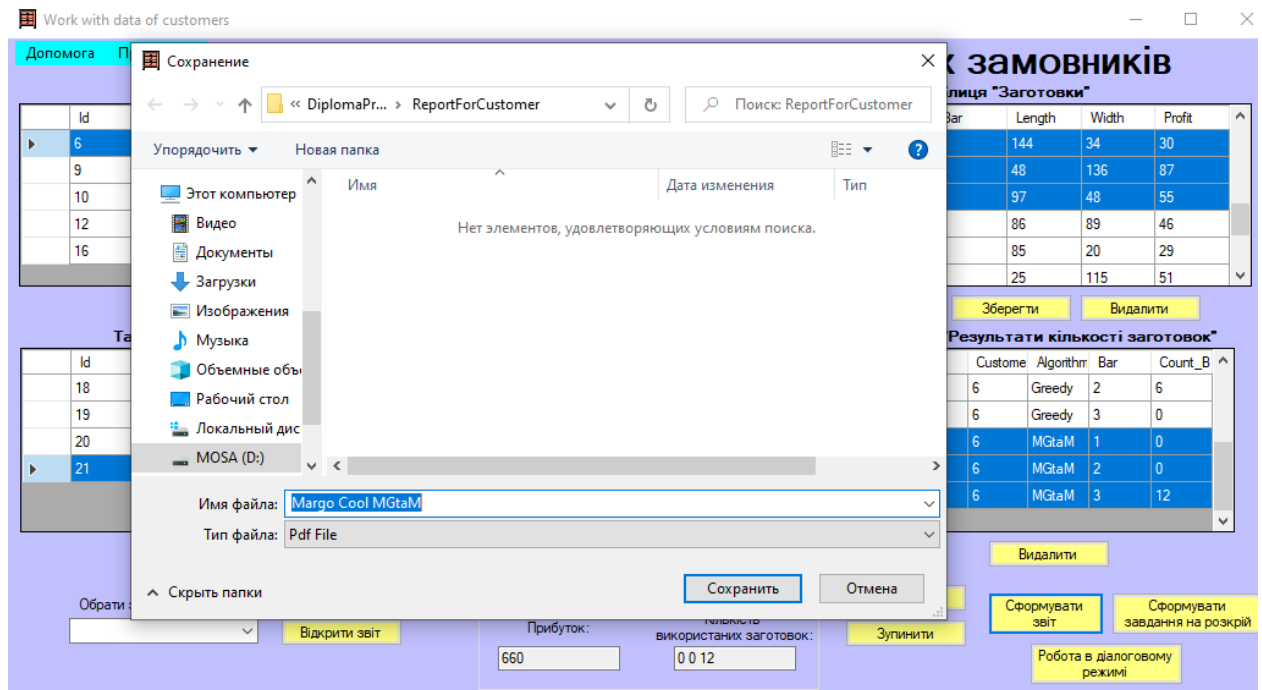


Рисунок 5.13 – Вікно збереження звіту

Автоматично назва файлу складається з ім'я та прізвища замовника та алгоритм, яким пораховано результат. Натискаємо «Зберегти». З'являється повідомлення про збереження. Щоб подивитися цей звіт обираємо у відповідному полі файл та натискаємо кнопку «Відкрити звіт». Звіт замовника зображено на рисунку 5.14.

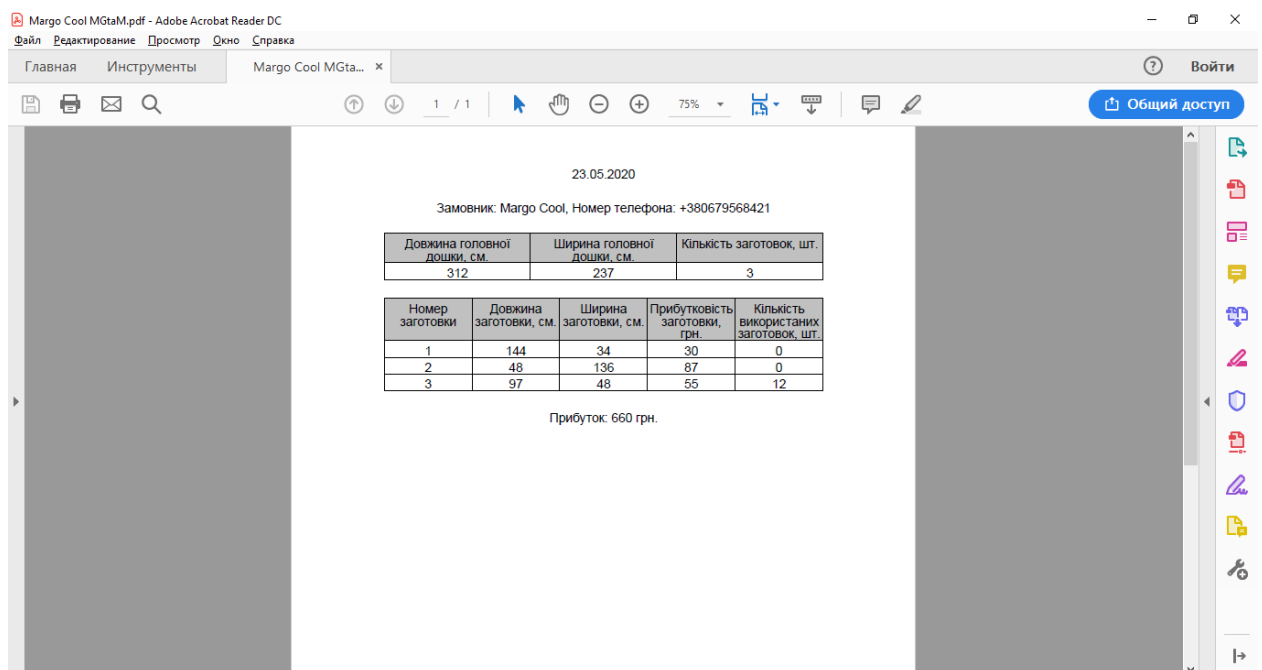


Рисунок 5.14 – Звіт для замовника Margo Cool

В цьому звіті наведені всі вхідні дані та вихідні, тобто результати розкрою, а також інформація про замовника та дата розрахунків.

Також можна сформувати завдання на розкрій, натиснувши кнопку «Сформувати завдання на розкрій». З'являється вікно для збереження, тільки папка за замовченням інша. Формат назви файлу такий самий як і для формування звіту. Вікно збереження для завдання на розкрій зображено на рисунку 5.15.

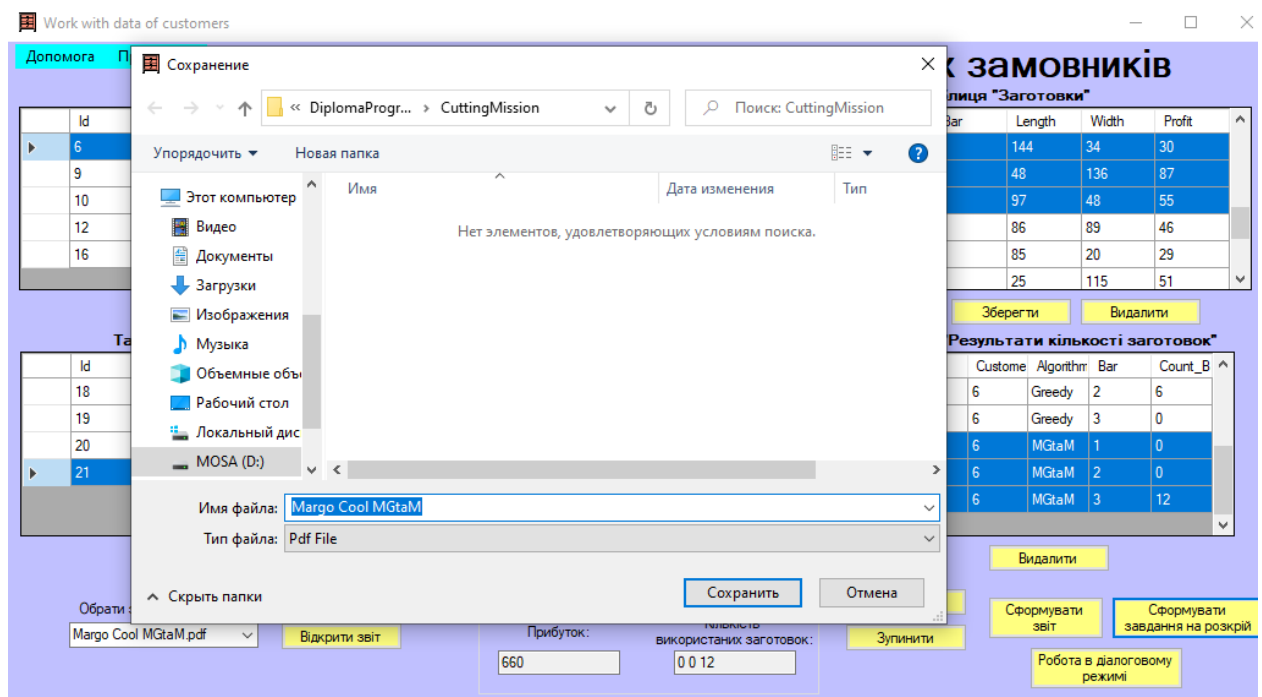


Рисунок 5.15 – Завдання на розкрій замовника Margo Cool

Натиснули «Зберегти». Потім у папці «CuttingMission» відкриваємо файл. Завдання на розкрій зображено на рисунку 5.16.

Margo Cool MGtaM.pdf - Adobe Acrobat Reader DC

Файл Редактирование Просмотр Окно Справка

Главная Инструменты Margo Cool MGta... x

Общий доступ

23.05.2020

Заказчик: Margo Cool, Номер телефона: +380679568421

Довжина головної дошки, см.	Ширина головної дошки, см.	Кількість заготовок, шт.
312	237	3

Номер заготовки	Довжина заготовки, см.	Ширина заготовки, см.	Кількість заготовок для розкрою, шт.
1	144	34	0
2	48	136	0
3	97	48	12

ПІБ виконавця: _____

Дата виконання: _____

Підпис: _____

Рисунок 5.16 – Завдання на розкрій

Так само тут знаходиться дата розрахунку, дані замовника, вхідні дані для розкрою та кількість заготовок необхідна для розкрою. А також поля, які необхідно заповнити розкрійнику для того, щоб було розуміння, що розкрій виконано та людина виконала свою роботу.

Далі можна проекспериментувати з даними. Для цього необхідно натиснути кнопку «Робота в діалоговому режимі». Відкриється вікно (те саме вікно, яке й відкриється при натисканні кнопки «Вибір варіанту розкрою в діалоговому режимі» на головній формі) для роботи в діалоговому режимі вже з сформованим планом розкрою, але дані можна буде змінювати, а якщо з головного вікна натиснути на кнопку – то поля будуть пустими. Вікно для експериментів наведено на рисунку 5.17.

Interactive mode

Допомога Про автора

Вибір варіанту розкрою в діалоговому режимі

Журнал результатів розкрою

	Id	Algorithm	Length_Ma	Width_Main	Count_Type	Vector_Len	Vector_Wid	Vector_Prol	Vector_Res	Income

Вхідні дані

Довжина головної дошки:

Ширина головної дошки:

Кількість заготовок:

Роздільник - пробіл
Довжини заготовок:

Ширини заготовок:

Прибутковість заготовок:

Введіть Id для занесення даних в базу даних:

Оберіть алгоритм для розрахунку:

Прибуток:

Кількість використаних заготовок:

Очистити поля

Обрати звіт для перегляду:

Рисунок 5.17 – Варіант розкрою в діалоговому режимі (вже дані занесені)

Змінюємо значення, наприклад, ширини головної дошки на 198, обираємо алгоритм «Жадібний алгоритм» та натискаємо «Порахувати». Бачимо результат. Потім спробуємо порахувати алгоритмом «Метод гілок та меж», також натискаємо «Порахувати» і бачимо результат. Якщо необхідно запинити підрахунок – натискаємо кнопку «Зупинити». Результат роботи алгоритмів зображено на рисунку 5.18.

Interactive mode

Допомога Про автора

Вибір варіанту розкрою в діалоговому режимі

Журнал результатів розкрою

	Id	Algorithm	Length_Ma	Width_Ma	Count_Type	Vector_Len	Vector_Wid	Vector_Prol	Vector_Res	Income
	1	Greedy	312	198	3	144 48 97	34 136 48	30 87 55	0 6 0	522
▶	2	MGtaM	312	198	3	144 48 97	34 136 48	30 87 55	0 0 12	660

Вхідні дані

Довжина головної дошки: 312

Ширина головної дошки: 198

Кількість заготовок: 3

Роздільник - пробіл

Довжини заготовок: 144 48 97

Ширини заготовок: 34 136 48

Прибутковість заготовок: 30 87 55

Введіть Id для занесення даних в базу даних: OK

Оберіть алгоритм для розрахунку:

Метод гілок та меж

Прибуток: 660

Кількість використаних заготовок: 0 0 12

Очистити поля

Обрати звіт для перегляду:

Відкрити звіт

Зупинити

Порахувати

Сформувати звіт

Сформувати завдання на розкрій

Рисунок 5.18 – Експерименти з вхідними даними

Бачимо, що результати занесені до «Журналу результатів розкрою». Так як не знаємо для кого робимо розрахунки, то відповідно даних замовника не має, а є лише вхідні та вихідні дані. Знов результати різні, краще результат дав алгоритм на основі методу гілок та меж. Кнопки «Сформувати звіт», «Відкрити звіт» та «Сформувати завдання на розкрій» роблять те саме, що й у вікні «Вибір варіанту розкрою на основі даних замовників», але для формування з'явиться вікно, де необхідно ввести дані замовника. Це саме вікно з'явиться і тоді, коли вирішимо занести один з варіантів розкрою у базу даних, для цього необхідно обрати поле, дані яких хочемо занести до бази даних та ввести у відповідне поле Id рядка, потім натиснути кнопку «ОК».

Вікно вводу даних замовника зображено на рисунку 5.19.

Interactive mode

Допомога Про автора

Вибір варіанту розкрою в діалоговому режимі

Журнал результатів розкрою

	Id	Algorithm	Length_Ma	Width_Mair	Count_Type	Vector_Len	Vector_Wid	Vector_Prof	Vector_Res	Income
▶	1	Greedy	312	198	3	144 48 97	34 136 48	30 87 55	0 6 0	522
	2	MGtaM	312	198	3	144 48 97	34 136 48	30 87 55	0 0 12	660

Data about the customer

Заповніть дані про замовника

Введіть ім'я:

Введіть прізвище:

Введіть номер телефона:

Вхідні дані:

Довжина:

Ширина головної дошки:

Кількість заготовок:

Вихідні дані:

Довжина:

Ширина заготовки:

Кількість заготовок:

Прибутковість заготовок:

Обрати звіт для перегляду:

Для занесення даних в базу даних:

Оберіть алгоритм для розрахунку:

Метод гілок та меж

Прибуток:

Кількість використаних заготовок:

Рисунок 5.19 – Введення даних замовника для збереження до бази даних

Натискаємо кнопку «Зберегти» та бачимо повідомлення про збереження. Закриваємо вікно та знову бачимо повідомлення про збереження. Можемо перейти до вікна «Вибір варіанту розкрою на основі даних замовників» та побачити, що дійсно дані збереглися. Результат зображено на рисунку 5.20.

Work with data of customers

Допомога Про автора **Вибір варіанту розкрою на основі даних замовників**

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
	9	Nastya	Bus	632322152	380	432	3
	10	Fedor	Car	935558825	362	174	2
	12	Nikita	Last	685462352	300	200	2
	16	Danilo	U	685642535	230	237	3
▶	18	Oleh	Hall	567833452	312	198	3

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
	36	16	2	48	136	87
	37	16	3	97	48	55
▶	38	18	1	144	34	30
	39	18	2	48	136	87
	40	18	3	97	48	55

Додати Зберегти Видалити Робота з таблицями

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
	18	1	Greedy	264
	19	1	MGtaM	264
	20	6	Greedy	522
	21	6	MGtaM	660
▶	22	18	Greedy	522

Введіть Id замовника для формування плану: 6 OK

Вхідні дані

Довжина головної дошки: 312

Роздільник - пробіл

Довжини заготовок: 144 48 97

Ширину головної дошки: 237

Ширини заготовок: 34 136 48

Кількість заготовок: 3

Прибутковість заготовок: 30 87 55

Таблиця "Результати кількості заготовок"

	Id	Custome	Algorithm	Bar	Count_B
	61	6	MGtaM	2	0
	62	6	MGtaM	3	12
▶	63	18	Greedy	1	0
	64	18	Greedy	2	6
	65	18	Greedy	3	0

Видалити

Оберіть алгоритм для розрахунку:

Метод гілок та меж

Прибуток: 660

Кількість використаних заготовок: 0 0 12

Обрати звіт для перегляду: Margo Cool MGtaM.pdf Відкрити звіт

Порахувати Сформувати звіт Сформувати завдання на розкрій

Зупинити Робота в діалоговому режимі

Рисунок 5.20 – Збереження даних для нового замовника

Повернемось на попереднє вікно. Також, якщо не відомо в якому порядку виконувати дії, натисніть кнопку «Допомога». З'явиться вікно інструкції користувача зображене на рисунку 5.21.

Help

Інструкція користувача

1. Введіть вхідні дані для формування плану розкрою у відповідні поля (блок "Вхідні дані").
2. Оберіть алгоритм у відповідному полі для підрахунку прибутку та кількості використаних заготовок.
 - 2.1. Жадібний алгоритм дає допустимий результат, але не обов'язково оптимальний.
 - 2.2. Метод гілок та меж (МГ та М) дає оптимальний результат, але, наприклад, при довжині головної дошки 2 м та заготовок 2 см знадобиться хвилин 10 для підрахунку.
3. Натисніть кнопку "Порахувати".
 - 3.1. Якщо всі дані коректні, то дочекайтесь результату.
 - 3.2. Якщо дані не коректні, то з'явиться відповідне повідомлення про помилку.
 - 3.3. Якщо бажаєте зупинити підрахунок натисніть кнопку "Зупинити".
4. Якщо бажаєте очистити поля натисніть кнопку "Очистити поля".
5. Передивіться результати у електронному журналі.
6. Якщо бажаєте занести дані в базу даних:
 - 6.1. Введіть Id рядка у відповідне поле.
 - 6.2. Якщо Id не існує, то з'явиться повідомлення про помилку.
 - 6.3. Натисніть кнопку "OK".
 - 6.4. З'явиться повідомлення про успішне збереження.
7. Якщо бажаєте сформувати звіт для замовника натисніть кнопку "Сформувати звіт".
8. Якщо бажаєте сформувати завдання на розкрій натисніть кнопку "Сформувати завдання на розкрій".
9. Якщо бажаєте передивитися звіт, оберіть звіт у відповідному полі та натисніть кнопку "Відкрити звіт".

Рисунок 5.21 – Вікно з інструкцією користувача для вікна «Вибір варіанту розкрою в діалоговому режимі»

Прочитавши інструкцію зачиняємо вікно натиском на «Хрест». Також у вікні можна подивитися інформацію про автора, як і у вікні «Вибір варіанту розкрою на основі даних замовників». Зачиняємо вікна натиском на «Хрест». Знову у головному вікні, для виходу натискаємо на кнопку «Вихід».

5.2 Випробування програмного продукту

Випробування програмного продукту буде здійснено шляхом тестування. Наведено опис тестів та порядок їх виконання. Тести проводяться для перевірки функціональних вимог, які наведені в технічному завданні для створення програмного продукту підтримки діяльності розкрійного цеху меблевого виробництва.

5.2.1 Мета випробувань

Мета випробувань – це перевірити відповідність функцій підтримки діяльності розкрійного цеху меблевого виробництва вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Перевіримо програмний продукт на відповідність функціональним вимогам. Випробування та їх результати наведені у таблицях 5.1 – 5.15.

Таблиця 5.1 – Випробування роботи жадібного алгоритму на основі введених даних

Випробування:	Робота жадібного алгоритму на основі введених даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою в діалоговому режимі»; введені дані в поля; обрано жадібний алгоритм
Дії:	Натиснути кнопку «Порахувати»
Очікуваний результат:	У полі «Прибуток» та «Кількість використаних заготовок» з'явиться результат роботи алгоритму
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.2 – Випробування роботи алгоритму на основі методу гілок та меж на основі введених даних

Випробування:	Робота алгоритму на основі методу гілок та меж на основі введених даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою в діалоговому режимі»; введені дані в поля; обрано алгоритм на основі методу гілок та меж
Дії:	Натиснути кнопку «Порахувати»
Очікуваний результат:	У полі «Прибуток» та «Кількість використаних заготовок» з'явиться результат роботи алгоритму
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.3 – Випробування збереження даних у базу даних на основі введених даних

Випробування:	Збереження даних у базу даних на основі введених даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою в діалоговому режимі»; в журналі результатів розкрою вже є дані; введено Id поля, якого дані будуть занесені до бази даних
Дії:	Натиснути кнопку «ОК»; ввести дані замовника у поля; натиснути кнопку «Зберегти»; закрити повідомлення; закрити вікно введення даних замовника
Очікуваний результат:	Побачити повідомлення про збереження даних у базу даних
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.4 – Випробування роботи жадібного алгоритму на основі наявних даних

Випробування:	Робота жадібного алгоритму на основі наявних даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; введено Id замовника; натиснута кнопка «ОК»; обрано жадібний алгоритм
Дії:	Натиснути кнопку «Порахувати»
Очікуваний результат:	У полі «Прибуток» та «Кількість використаних заготовок» з'явиться результат роботи алгоритму
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.5 – Випробування роботи алгоритму на основі методу гілок та меж на основі наявних даних

Випробування:	Робота алгоритму на основі методу гілок та меж на основі наявних даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; введено Id замовника; натиснута кнопка «ОК»; обрано алгоритм на основі методу гілок та меж
Дії:	Натиснути кнопку «Порахувати»
Очікуваний результат:	У полі «Прибуток» та «Кількість використаних заготовок» з'явиться результат роботи алгоритму
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.6 – Випробування видалення даних з бази даних на основі наявних даних

Випробування:	Видалення даних з бази даних на основі наявних даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; в таблиці «Результати доходу» чи «Результати кількості заготовок» вже є дані; обрано поле в таблиці
Дії:	Натиснути кнопку «Видалити»
Очікуваний результат:	Пропаде поле в таблиці
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.7 – Випробування додавання даних замовника в базу даних

Випробування:	Додавання даних замовника в базу даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; натиснута кнопка «Додати» під таблицею «Замовники»; заповнені поля нового замовника
Дії:	Натиснути кнопку «Зберегти» під таблицею «Замовники»
Очікуваний результат:	В новому полі з'явиться Id замовника
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.8 – Випробування видалення даних замовника з бази даних

Випробування:	Видалення даних замовника з бази даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; обрано поле в таблиці «Замовники»; натиснута кнопка «Видалити» під таблицею «Замовники»
Дії:	Натиснути кнопку «Зберегти» під таблицею «Замовники»
Очікуваний результат:	В таблиці «Замовники» не має видаленого поля
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.9 – Випробування змінення даних замовника в базі даних

Випробування:	Змінення даних замовника в базі даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; двічі натиснуте поле в таблиці «Замовники»; змінено дані в полі
Дії:	Натиснути кнопку «Зберегти» під таблицею «Заготовки»
Очікуваний результат:	В таблиці «Замовники» в обраному полі змінено значення
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.10 – Випробування додавання даних заготовки в базу даних

Випробування:	Додавання даних заготовки в базу даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; натиснута кнопка «Додати» під таблицею «Заготовки»; заповнені поля нової заготовки
Дії:	Натиснути кнопку «Зберегти» під таблицею «Заготовки»
Очікуваний результат:	В новому полі з'явиться Id заготовки
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.11 – Випробування видалення даних заготовки з бази даних

Випробування:	Видалення даних заготовки з бази даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; обрано поле в таблиці «Заготовки»; натиснута кнопка «Видалити» під таблицею «Заготовки»
Дії:	Натиснути кнопку «Зберегти» під таблицею «Заготовки»
Очікуваний результат:	В таблиці «Заготовки» не має видаленого поля
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.12 – Випробування змінення даних заготовки в базі даних

Випробування:	Змінення даних заготовки в базі даних
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; натиснута кнопка «Робота з таблицями»; двічі натиснуте поле в таблиці «Заготовки»; змінено дані в полі
Дії:	Натиснути кнопку «Зберегти» під таблицею «Заготовки»
Очікуваний результат:	В таблиці «Заготовки» в обраному полі змінено значення
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.13 – Випробування формування звіту для замовника

Випробування:	Формування звіту для замовника
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; введено Ід замовника; натиснута кнопка «ОК»; обрано будь-який алгоритм; натиснута кнопка «Порахувати»
Дії:	Натиснути кнопку «Сформувати звіт»; натиснути кнопку «Зберегти»
Очікуваний результат:	Повідомлення про успішне збереження файлу
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.14 – Випробування формування завдання на розкрій

Випробування:	Формування завдання на розкрій
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; введено Іd замовника; натиснута кнопка «ОК»; обрано будь-який алгоритм; натиснута кнопка «Порахувати»
Дії:	Натиснути кнопку «Сформувати завдання на розкрій»; натиснути кнопку «Зберегти»
Очікуваний результат:	Повідомлення про успішне збереження файлу
Очікуваний результат співпадає з отриманим:	Так

Таблиця 5.15 – Випробування перегляд звіту замовника

Випробування:	Перегляд звіту замовника
Початковий стан:	Відкрито вікно «Вибір варіанту розкрою на основі даних замовників(з бази даних)»; обрано звіт у полі «Обрати звіт для перегляду»
Дії:	Натиснути кнопку «Відкрити звіт»
Очікуваний результат:	Звіт відкрито для перегляду
Очікуваний результат співпадає з отриманим:	Так

Висновок до розділу

У розділі «Технологічний розділ» описано керівництво користувача та випробування програмного продукту.

Детально описано та зображено керівництво користувача, тобто поетапно зображено користування програмним продуктом та детально описано всі дії.

Наведено мету випробувань, загальні положення та результати випробувань. Результати випробувань виконувались відповідно до

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

функціональних вимог та наведені в таблицях. Всі випробування успішно пройдені.

ЗАГАЛЬНІ ВИСНОВКИ

Даний дипломний проєкт розроблено для підтримки діяльності розкрійного цеху меблевого виробництва.

У розділі «Загальні положення» було розглянуто предметно середовище, аналоги існуючих систем, описано функціональну модель, процес діяльності та призначення розробки. Також сформована мета та задачі, які допомагають досягти поставленої мети. Схема структурна діяльності поділена на три сектори: актор, вхідна/вихідна інформація та діяльність. Актором виступає менеджер. Описано актора системи та його варіанти використання для схеми структурної варіантів використання. Метою є підвищення ефективності виробничого процесу розкрійного цеху меблевого виробництва.

У розділі «Інформаційне забезпечення» було описано вхідні та вихідні дані, а також структуру бази даних. Для вхідних даних наведено не тільки обмеження типів даних, а й логічні обмеження. Для структури бази даних детально описано таблиці та зв'язки між ними.

В математичному забезпеченні описано змістовну постановку задачі, математичну поставку задачі розкрою, яка є базовою задачею для діяльності розкрійного цеху меблевого виробництва, обґрунтовано та описано методи розв'язання поставленої задачі. Задача є NP-складною, тому обрано два алгоритму для її вирішення: жадібний, який легкий у реалізації, дає допустимий розв'язок і алгоритм на основі методу гілок та меж, який дає оптимальний розв'язок.

У розділі «Програмне та технічне забезпечення» описано засоби розробки, вимоги до технічного забезпечення, архітектуру програмного забезпечення та звіти. Обрані засоби розробки створені компанією Microsoft, тому гарно поєднуються. Для розуміння архітектури програмного забезпечення наведені та описані діаграми класів, послідовності та компонентів.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

В технологічному розділі описано керівництво користувача та випробування програмного продукту. Керівництво користувача детально описано, що дає можливість легко користуватися програмним продуктом. Всі випробування пройдені успішно та відповідають функціональним вимогам.

Розроблений програмний продукт підвищує ефективність розкрійного цеху меблевого виробництва за рахунок підвищення прибутку та зменшення часу на розкрій.

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

ПЕРЕЛІК ПОСИЛАНЬ

1. Алексеева Е. В. Построение математических моделей целочисленного линейного программирования. Примеры и задачи: Учеб. пособие / Е. В. Алексеева. – Новосибирск: НГУ, 2012. – 131 с.
2. Левитин А.В. Алгоритмы: введение в разработку и анализ: пер. с англ. / А.В. Левитин. – М.: Издательский дом «Вильямс», 2006. – 576 с.
3. Ахо А.В. Структуры данных и алгоритмы: пер. с англ.: уч. пос. / А.В. Ахо, Дж. Хопкрофт, Дж.Д. Ульман. – М.: Издательский дом «Вильямс», 2007. – 400 с.
4. Булавский В. А. Численные методы линейного программирования (специальные задачи). / В. А. Булавский, Р. А. Звягина, М. А. Яковлева. – М.: Наука, 1977. – 368 с.
5. Хохлюк В. И. Задачи целочисленной оптимизации (алгоритмы): Учеб. пособие. / В. И. Хохлюк. – Новосибирск: НГУ, 1980. – 92 с.
6. Банди Б. Методы оптимизации. / Б. Банди. – М.: Радио и связь, 1988. – 128 с.
7. Пападимитриу Х. Комбинаторная оптимизация. Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. - М.: Мир, 1984. – 510 с.
8. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации / И.В. Сергиенко. – К.: Наук. думка, 1988. – 471 с.
9. Сергиенко И.В. Задачи дискретной оптимизации: проблемы, методы решения, исследования / И.В. Сергиенко, В.П. Шило. – Киев: Наук. думка, 2003. – 264 с.
10. Схрейвер А. Теория линейного и целочисленного программирования / А. Схрейвер. В двух томах. – М.: Мир, 1991. Т. 1. – 362 с.; Т. 2. – С. 363 – 704.

11. Microsoft Visual Studio [Электронный ресурс] // Wikipedia. – 2020. – Режим доступа до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio.
12. Лучшие IDE для разработки на C# [Электронный ресурс] // GeekBrains. – 2020. – Режим доступа до ресурсу: https://geekbrains.ru/posts/c_sharp_ides.
13. Eclipse Java: среда разработки “под себя” [Электронный ресурс] // JavaRush. – 2020. – Режим доступа до ресурсу: <https://javarush.ru/groups/posts/2359-obzor-eclipse-java-sreda-razrabotki-pod-sebja>.
14. Введение в C# [Электронный ресурс] // Metanit. – 2020. – Режим доступа до ресурсу: <https://metanit.com/sharp/tutorial/1.1.php>.
15. C# – Преимущества и недостатки [Электронный ресурс] // ThemeIsle. – 2020. – Режим доступа до ресурсу: <https://shwanoff.ru/plus-minus-c-sharp>.
16. C# - Энциклопедия языков программирования [Электронный ресурс] // Progopedia. – 2020. – Режим доступа до ресурсу: <http://progopedia.ru/language/csharp>.
17. Введение в Java [Электронный ресурс] // Metanit. – 2020. – Режим доступа до ресурсу: <https://metanit.com/java/tutorial/1.1.php>.
18. Плюсы и минусы программирования на Java [Электронный ресурс] // Medium. – 2020. – Режим доступа до ресурсу: <https://medium.com/nuances-of-programming/плюсы-и-минусы-программирования-на-java-2861f4c2a0d5>.
19. Microsoft SQL Server [Электронный ресурс] // Wikipedia. – 2020. – Режим доступа до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_SQL_Server.
20. Сравнение современных СУБД [Электронный ресурс] // Drach. – 2020. – Режим доступа до ресурсу: <http://drach.pro/blog/hi-tech/item/145-db-comparison>.

21. MongoDB [Электронный ресурс] // Wikipedia. – 2020. – Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/MongoDB>.
22. Рассуждение на тему, какую базу данных выбирать [Электронный ресурс] // ТМ. – 2020. – Режим доступа до ресурсу: <https://habr.com/ru/post/348220>.

Додаток А

Тексти програмного коду

**Інформаційна система підтримки діяльності розкрійного цеху
меблевого виробництва**

(Найменування програми (документа))

DVD-R

(Вид носія даних)

10 арк, 23300 Кб

(Обсяг програми (документа), арк.,) Кб)

Київ – 2020 року

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DiplomaProgram
{
    public class Workpiece
    {
        public int Width { get; set; }
        public int Length { get; set; }
        public int Price { get; set; }
        public double Utilty { get; set; }
        public int Quantity { get; set; }
        public Workpiece(int lenght, int width, int price)
        {
            Width = width;
            Length = lenght;
            Price = price;
        }
        public Workpiece()
        {
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DiplomaProgram
{
    public class BarsDB
    {
        public int PositionBar { get; set; }
        public int NumberBar { get; set; }
        public int Length { get; set; }
        public int Width { get; set; }
        public int Profit { get; set; }
        public BarsDB(int positionBar, int numberBar, int lenght, int width, int profit)
        {
            PositionBar = positionBar;
            NumberBar = numberBar;
            Length = lenght;
            Width = width;
            Profit = profit;
        }
        public BarsDB()
    }
}
```



```

    {
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiplomaProgram
{
    public partial class DataCustomer : Form
    {
        UsualWork usualwork;
        ToolTip tt1, tt2, tt3, tt4;
        int counter=0;
        public DataCustomer()
        {
            InitializeComponent();
            tt1 = tt2 = tt3 = tt4 = new ToolTip();
            tt1.SetToolTip(namebox, "Введіть ім'я замовника");
            tt2.SetToolTip(surnamebox, "Введіть прізвище замовника");
            tt3.SetToolTip(numberbox, "Введіть номер телефона замовника");
            tt4.SetToolTip(savebtn, "Натисніть кнопку для збереження даних");
            tt1.IsBalloon = tt2.IsBalloon = tt3.IsBalloon = tt4.IsBalloon = true;
        }
        public DataCustomer(UsualWork uw)
        {
            InitializeComponent();
            usualwork=uw;
            tt1 = tt2 = tt3 = tt4 = new ToolTip();
            tt1.SetToolTip(namebox, "Введіть ім'я замовника");
            tt2.SetToolTip(surnamebox, "Введіть прізвище замовника");
            tt3.SetToolTip(numberbox, "Введіть номер телефона замовника");
            tt4.SetToolTip(savebtn, "Натисніть кнопку для збереження даних");
            tt1.IsBalloon = tt2.IsBalloon = tt3.IsBalloon = tt4.IsBalloon = true;
        }
        private void namebox_KeyPress(object sender, KeyPressEventArgs e)
        {
            char number = e.KeyChar;
            if (!Char.IsLetter(number) && number != 8)
            {
                e.Handled = true;
            }
        }
    }
}

```

```

private void surnamebox_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsLetter(number) && number != 8)
    {
        e.Handled = true;
    }
}
private void numberbox_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8)
    {
        e.Handled = true;
    }
}
private void savebtn_Click(object sender, EventArgs e)
{
    if (namebox.Text != "" && surnamebox.Text != "" && numberbox.Text != "")
    {
        usualwork.nameCustomer = namebox.Text;
        usualwork.surnameCustomer = surnamebox.Text;
        usualwork.phoneCustomer = Convert.ToInt32(numberbox.Text);
        counter = 1;
        MessageBox.Show("Saved!");
    }
    else MessageBox.Show("Enter data of customer!");
}
private void DataCustomer_FormClosing(object sender, FormClosingEventArgs e)
{
    if (counter == 0)
    {
        MessageBox.Show("Enter data of customer!");
        e.Cancel = true;
    }
}
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Timers;

namespace DiplomaProgram
{
    public class GreedyAlgoritm
    {
        int Sum = 0;
        List<Workpiece> workpieces;
    }
}

```

```

public GreedyAlgorithm(List<Workpiece> wp)
{
    workpieces = wp;
}
public void UtiltyCalc(List<Workpiece> workpieces)
{
    for (int i = 0; i < workpieces.Count; i++)
        workpieces[i].Utilty = Convert.ToDouble(workpieces[i].Price) /
        Convert.ToDouble(workpieces[i].Width * workpieces[i].Length);
}
public int Work(int widthB, int length, List<Workpiece> workpieces)
{
    int minW = workpieces.Select(x => x.Width).Min();
    int minL = workpieces.Select(x => x.Length).Min();
    List<Workpiece> sortedworkpieces = workpieces.OrderByDescending(o
    => o.Utilty).ToList();
    int i = 0;
    do
    {
        if (sortedworkpieces[i].Length <= length)
        {
            length = length - sortedworkpieces[i].Length;
            int j = 0;
            int width = widthB;
            do
            {
                if (sortedworkpieces[j].Width <= width && sortedworkpieces[j].Length ==
                sortedworkpieces[i].Length)
                {
                    width = width - sortedworkpieces[j].Width;
                    Sum = Sum + sortedworkpieces[j].Price;
                    sortedworkpieces[j].Quantity++;
                }
                else j++;
            } while (width >= minW && (j) < sortedworkpieces.Count);
        }
        else i++;
    } while (length >= minL && i < sortedworkpieces.Count);
    workpieces = sortedworkpieces;
    return Sum;
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;
using System.Runtime.Serialization;
using System.Threading;

namespace DiplomaProgram
{
    public class Node
    {
        public int Length { get; set; }
        public int Mark { get; set; }
        public int AddedValue { get; set; }
        public List<Workpiece> wp { get; set; }
        public Node(int length, int mark, int addedvalue)
        {
            Length = length;
            Mark = mark;
            AddedValue = addedvalue;
        }
        public Node()
        {
        }
    }
    public class MGtaM
    {
        CancellationToken token;
        List<Node> Nodes = new List<Node>();
        public void CreateNode(int lenght, int mark, int addedvalue)
        {
            Nodes.Add(new Node(lenght, mark, addedvalue));
        }
        public int Record_value { get; set; }
        int Sum = 0;
        int minW, minL;
        List<Workpiece> workpieces;
        List<Workpiece> sortedworkpieces;
        public Node RecordNode { set; get; }
        public MGtaM(List<Workpiece> wp, CancellationToken tkn)
        {
            workpieces = wp;
            token = tkn;
        }
        public void UtiltyCalc(List<Workpiece> workpieces)
        {
            for (int i = 0; i < workpieces.Count; i++)
                workpieces[i].Utilty = Convert.ToDouble(workpieces[i].Price) /
                Convert.ToDouble(workpieces[i].Width * workpieces[i].Length);
        }
    }
}

```

					ДП 6316.00.000 ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        minW = workpieces.Select(x => x.Width).Min();
        minL = workpieces.Select(x => x.Length).Min();
    }
    static int Max(int a, int b)
    {
        if (a >= b)
            return a;
        else return b;
    }
    public int GreedyComponent(int length_left, int width)
    {
        Sum = 0;
        int minW = workpieces.Select(x => x.Width).Min();
        sortedworkpieces = workpieces.OrderByDescending(o => o.Utily).ToList();
        int i = 0;
        do
        {
            if (sortedworkpieces[i].Width <= width && sortedworkpieces[i].Length == length_left)
            {
                width = width - sortedworkpieces[i].Width;
                Sum = Sum + sortedworkpieces[i].Price;
                sortedworkpieces[i].Quantity++;
            }
            else i++;
        } while (width >= minW && (i < sortedworkpieces.Count));
        return Sum;
    }
    public void AddingValues(Node oldnode, Node newnode)
    {
        for (int i = 0; i < newnode.wp.Count; i++)
        {
            newnode.wp[i].Quantity = oldnode.wp[i].Quantity + sortedworkpieces[i].Quantity;
            sortedworkpieces[i].Quantity = 0;
        }
    }
    public void AlgoritmWorking(List<Workpiece> workpieces, int width, Node node)
    {
        if (token.IsCancellationRequested)
        {
            return;
        }
        sortedworkpieces = workpieces.OrderByDescending(o => o.Utily).ToList();
        List<Workpiece> listwithdeleted = sortedworkpieces.GroupBy(p => p.Length).Select(g
=> g.First()).ToList();

        for (int i = 0; i < listwithdeleted.Count; i++)
        {
            if (node.Mark > Record_value)
            {
                int length_left = node.Length - listwithdeleted[i].Length;
                if (length_left >= 0)

```

```

        {
            int exact_part_of_mark = GreedyComponent(listwithdeleted[i].Length, width);

            int mark = node.AddedValue + exact_part_of_mark + (length_left * width) /
(listwithdeleted[0].Length * listwithdeleted[0].Width) * listwithdeleted[0].Price;
            Node newnode = new Node(length_left, mark, node.AddedValue +
exact_part_of_mark);
            newnode.wp = new List<Workpiece>();
            foreach (var item in node.wp)
            {
                newnode.wp.Add(new Workpiece
                {
                    Width = item.Width,
                    Length = item.Length,
                    Price = item.Price,
                    Utilty = item.Utilty,
                    Quantity = item.Quantity
                });
            }
            AddingValues(node, newnode);
            Nodes.Add(newnode);
        }
        else
        {
            if (node.AddedValue > Record_value)
            {
                Record_value = Max(node.AddedValue, Record_value);
                RecordNode = new Node { AddedValue = node.AddedValue, Length =
node.Length, Mark = node.Mark, wp = node.wp };
            }
        }
    }
    else
    {
        Nodes.Remove(node);
        break;
    }
}
if (node != null)
    Nodes.Remove(node);
if (Nodes.Count != 0)
{
    Node nodeBest = Nodes.Find(x => x.Mark == Nodes.Select(v => v.Mark).Max());
    AlgoritmWorking(listwithdeleted, width, nodeBest);
}
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiplomaProgram
{
    public partial class Author : Form
    {
        public Author()
        {
            InitializeComponent();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiplomaProgram
{
    public partial class HelpUsual : Form
    {
        public HelpUsual()
        {
            InitializeComponent();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiplomaProgram
{
    public partial class HelpBD : Form
    {

```

					ДП 6316.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

        public HelpBD()
        {
            InitializeComponent();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiplomaProgram
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            foreach (Form f in Application.OpenForms)
            {
                if (f.Name == "BDWork")
                {
                    MessageBox.Show("It has already opened!");
                    return;
                }
            }
            BDWork bdwork = new BDWork();
            if(bdwork.clProg == 0)
            bdwork.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            foreach (Form f in Application.OpenForms)
            {
                if (f.Name == "UsualWork")
                {

```



```
        MessageBox.Show("It has already opened!");
        return;
    }
}
UsualWork usualwork = new UsualWork();
usualwork.Show();
}
}
```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проєкту

_____ Оксана ЖУРАКОВСЬКА

(підпис)

(вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

(підпис)

(вл. ім'я, прізвище)

“14” квітня 2020 р.

Інформаційна система підтримки діяльності розкрійного цеху
меблевого виробництва

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6316.01.000 ТЗ*

на 11 сторінках

Київ – 2020 року

ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
1.1	Повне найменування системи та її умовне позначення	3
1.2	Найменування організації-замовника та організацій-учасників робіт .	3
1.3	Перелік документів, на підставі яких створюється система	3
1.4	Планові терміни початку і закінчення роботи зі створення системи....	4
2	ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	5
2.1	Призначення системи	5
2.2	Цілі створення системи	5
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	6
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	7
4.1	Вимоги до функціональних характеристик	7
4.2	Вимоги до надійності	8
4.3	Вимоги до складу і параметрів технічних засобів	8
5	СТАДІЇ І ЕТАПИ РОЗРОБКИ	10
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	11
6.1	Види випробувань	11

					ДП 6316.01.000 ТЗ			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розроб.		Мосьяк О.І.			Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва	Лім.	Лист	Листів
Перевірів.		Жураковська О.С.					2	11
Н. кон.		Проскура С.Л.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63		
Затв.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повне найменування системи: Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва.

Умовне позначення: «Cutting Shop».

1.2 Найменування організації-замовника та організацій-учасників робіт

Організація-замовник є Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», факультет інформатики та обчислювальної техніки, кафедра автоматизованих систем обробки інформації та управління. Представником замовника є Жураковська Оксана Сергіївна. Адреса замовника: м. Київ, в. Політехнічна 41, корпус 18.

Організація-учасник — студент групи ІС-63 Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського», факультету інформатики та обчислювальної техніки, кафедри автоматизованих систем обробки інформації та управління Мосьпак Олег Ігорович.

1.3 Перелік документів, на підставі яких створюється система

Документи, на підставі яких створюється система та документація до неї:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;

- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи зі створення системи «Cutting Shop» – 13 квітня 2020 року.

Плановий термін закінчення роботи зі створення системи «Cutting Shop» – не пізніше 1 червня 2020 року.

					ДП 6316.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Призначенням розробки є підтримка діяльності розкрійного цеху меблевого виробництва.

2.2 Цілі створення системи

Метою є підвищення ефективності виробничого процесу розкрійного цеху меблевого виробництва.

Щоб досягти поставлену мету, потрібно вирішити наступні задачі:

- проведення аналізу алгоритмів, які підходять до поставленої задачі;
- розробка алгоритмів вирішення задачі розкрою для максимізації прибутку;
- ведення замовників;
- ведення заготовок замовників;
- формування планів розкрою;
- створення програмного застосування;
- формування звітів;
- формування завдання на розкрій.

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є процес виробництва та взаємодії з клієнтами для розкрійного цеху.

Працювати з програмним продуктом можуть всі користувачі, які мають доступ до застосунку та доступ до бази даних. За допомогою застосунку користувач може вести роботу з базою даних, знаходити максимальний прибуток для замовника та сформулювати звіт для нього, а також сформулювати завдання на розкрій.

					ДП 6316.01.000 ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Програмний продукт повинен виконувати наступні функціональні вимоги до системи:

- надання можливості користувачу сформувати план розкрою на основі введених даних:
 - 1) обрати алгоритм для розрахунків;
 - 2) побачити результати розрахунків;
 - 3) додати результат у базу даних;
- надання можливості користувачу сформувати план розкрою на основі наявних даних:
 - 1) обрати алгоритм для розрахунків;
 - 2) побачити результати розрахунків;
 - 3) видалити результат з бази даних;
- надання можливості користувачу ведення замовників:
 - 1) додавати дані замовників до бази даних;
 - 2) видаляти дані замовників до бази даних;
 - 3) змінювати дані замовників у базі даних;
- надання можливості користувачу ведення заготовок:
 - 1) додавати дані заготовок до бази даних;
 - 2) видаляти дані заготовок до бази даних;
 - 3) змінювати дані заготовок у базі даних;
- надання можливості користувачу сформувати звіт для замовника;
- надання можливості користувачу сформувати завдання на розкрій;
- надання можливості користувачу переглянути звіт замовника.

4.2 Вимоги до надійності

Програмний продукт при виникненні помилок повинен повідомити про це та продовжити свою роботу. Помилки можуть стосуватися:

- збою зв'язку з базою даних;
- не вірному вводі даних у базу даних;
- не відповідності вхідних даних;
- не відповідності алгоритму дій.

Програма виконуючи функціональні вимоги повинна забезпечити надійність, через це при виникненні помилки, програмний продукт повідомить про неї та дасть підказку для правильних дій. При правильному користуванні програмним продуктом ніяких аварійних ситуацій не повинно виникати.

4.3 Вимоги до складу і параметрів технічних засобів

Для вірного функціонування програмного продукту необхідно такі технічні засоби:

- комп'ютер з такими характеристиками:
 - 2) об'єм оперативної пам'яті не менше 512 МБ;
 - 3) процесор з тактовою частотою не нижче 1 ГГц;
 - 4) інші характеристики суттєво не впливають на роботу програмного продукту;
- наявне програмне забезпечення:
 - 1) операційна система Windows 7/8/10;
 - 2) база даних SQL Server;
 - 3) застосунок SQL Server 2014 Management Studio і вище;
 - 4) NET Framework 3.5 і вище;
- комп'ютерна периферія:
 - 1) монітор;

2) клавіатура;

3) мишка.

					ДП 6316.01.000 ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

У таблиці 5.1 наведено календарний план робіт та терміни виконання.

Таблиця 5.1 – Календарний план виконання робіт

№ п/п	Назва етапу роботи	Термін виконання етапу
1.	Підготовка технічного завдання	13.04.2020
2.	Опис процесу діяльності та функціональної моделі	16.04.2020
3.	Огляд аналогів програмного продукту	18.04.2020
4.	Розробка інформаційного забезпечення	22.04.2020
5.	Розробка математичного забезпечення	26.04.2020
6.	Розробка програмного забезпечення	30.04.2020
7.	Створення програмного продукту	10.05.2020
8.	Узгодження з керівником програмного продукту	11.05.2020
9.	Випробування програмного продукту	13.05.2020
10.	Здача готового програмного продукту	14.05.2020

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

6.1 Види випробувань

Для перевірки роботи програмного продукту буде проведено випробування, узгоджене з замовником. Мета випробування: перевірити основні функції системи. Випробування, які перевіряли програмний продукт та методика тестувань наведено у пояснювальній записці. Здача програмного продукту відбудеться на комп'ютері виконавця завдання з встановленим програмним забезпеченням, що відповідає вимогам.

					ДП 6316.01.000 ТЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:
Попенко Володимир Дмитрович

ID перевірки:
1003783782

Дата перевірки:
04.06.2020 18:35:11 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2020 20:18:53 EEST

ID користувача:
77149

Назва документу: Mospak_bachelor

ID файлу: 1003798227 Кількість сторінок: 67 Кількість слів: 8964 Кількість символів: 65570 Розмір файлу: 1.44 MB

4.69% Схожість

Найбільша схожість: 2.26% з джерело бібліотеки. ID файлу: 5859077

3.88% Схожість з Інтернет джерелами

19

Page 69

4.18% Текстові збіги по Бібліотеці акаунту

114

Page 69

0.11% Цитат

Цитати

1

Page 70

Вилучення переліку посилань вимкнено

0% Вилучень

Вилучений текст відсутній

Підміна символів

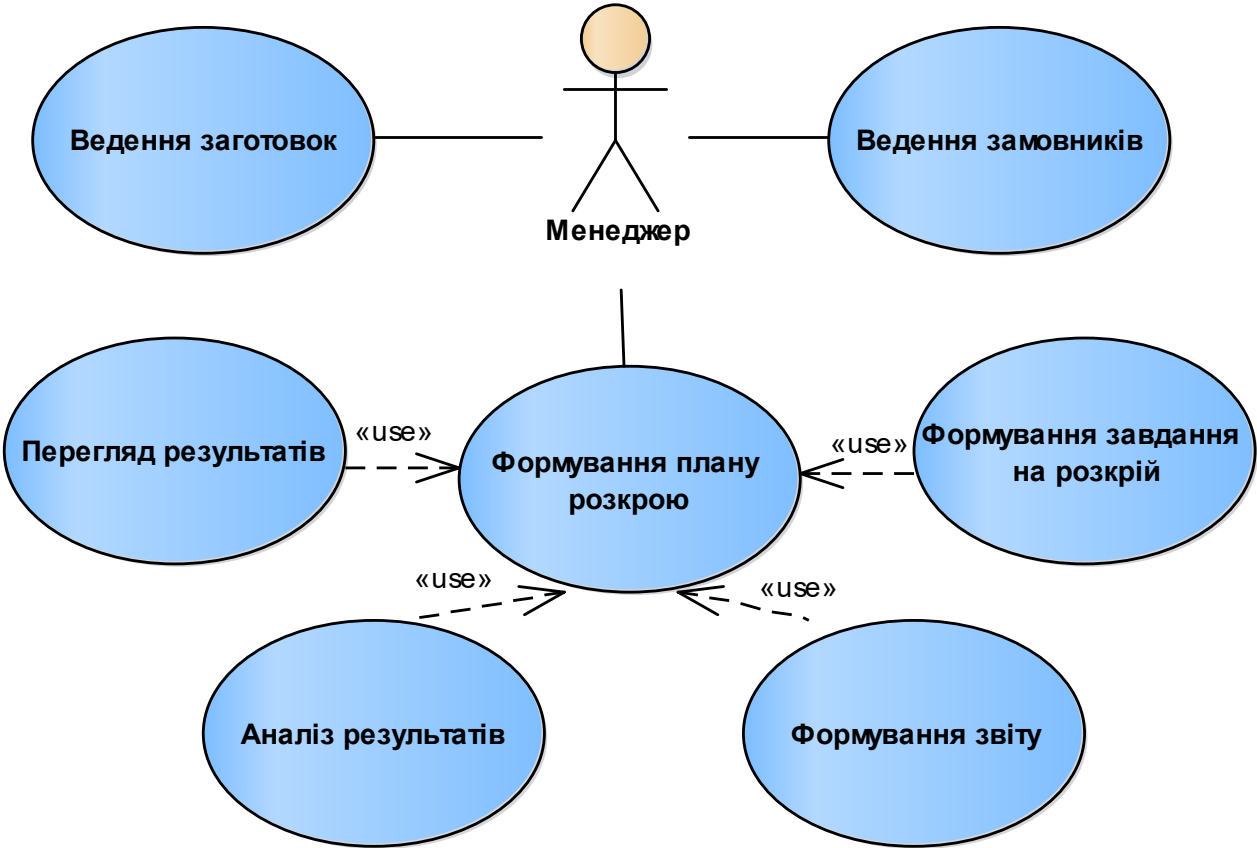
Заміна символів

2

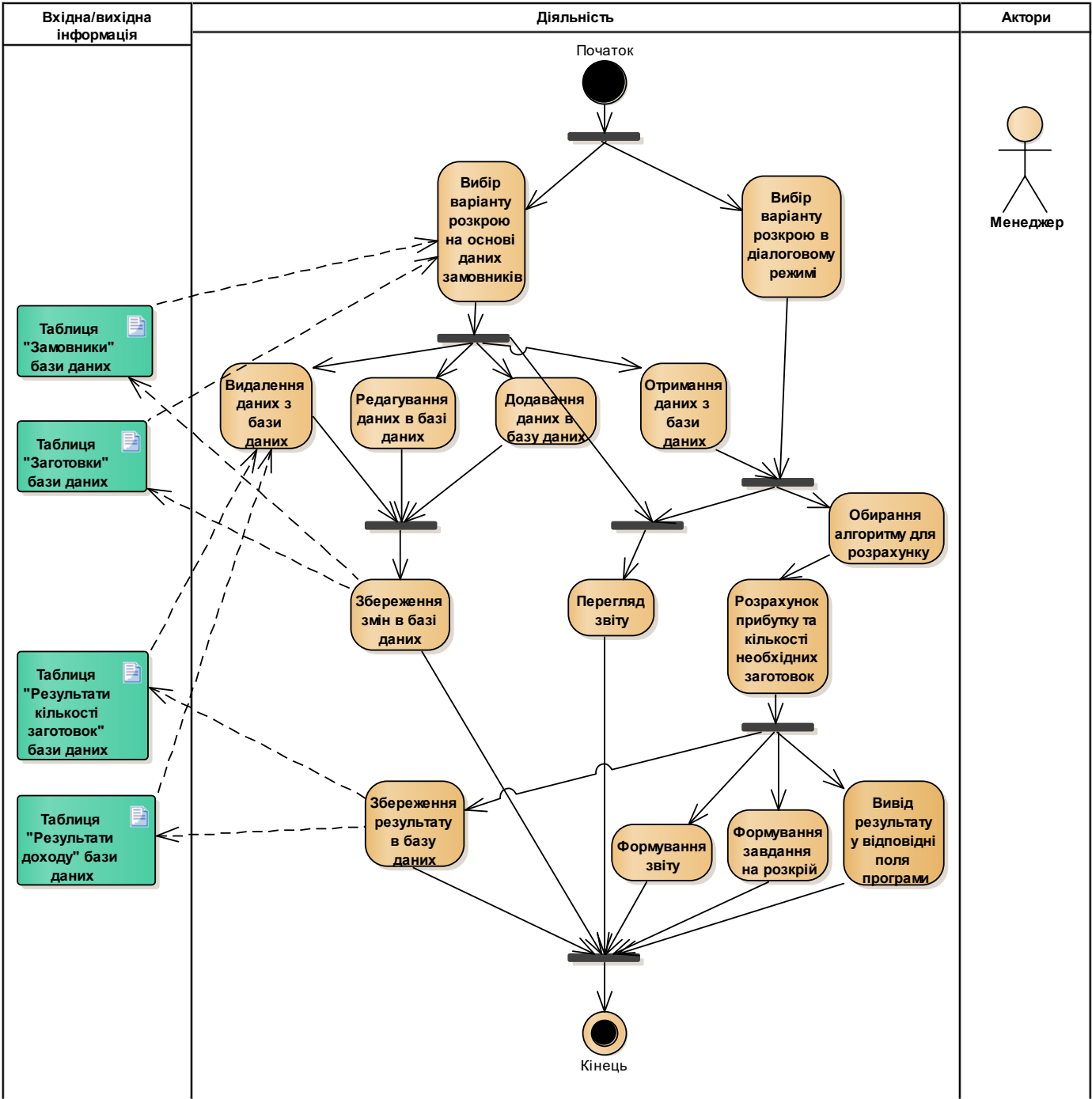
Графічний матеріал до дипломного проєкту

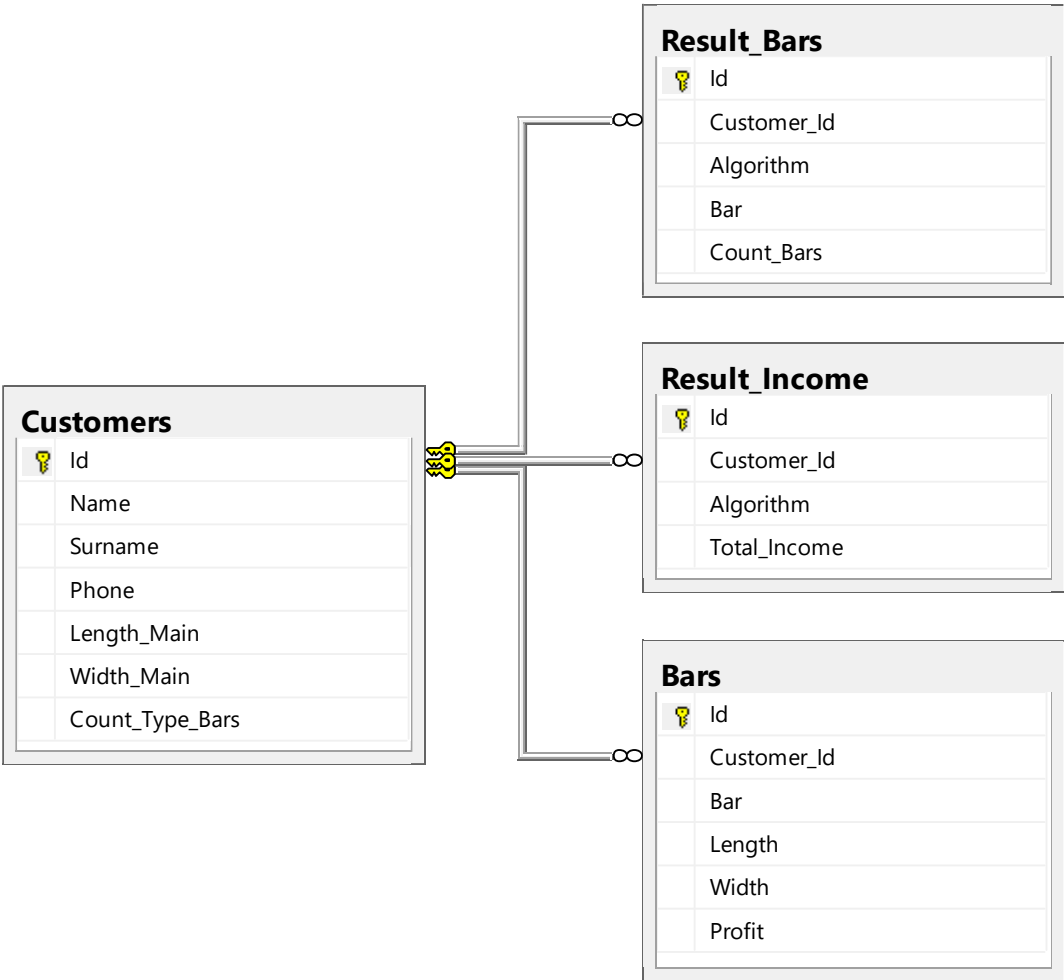
на тему: Інформаційна система підтримки діяльності
розкрийного цеху меблевого виробництва

Київ – 2020 року

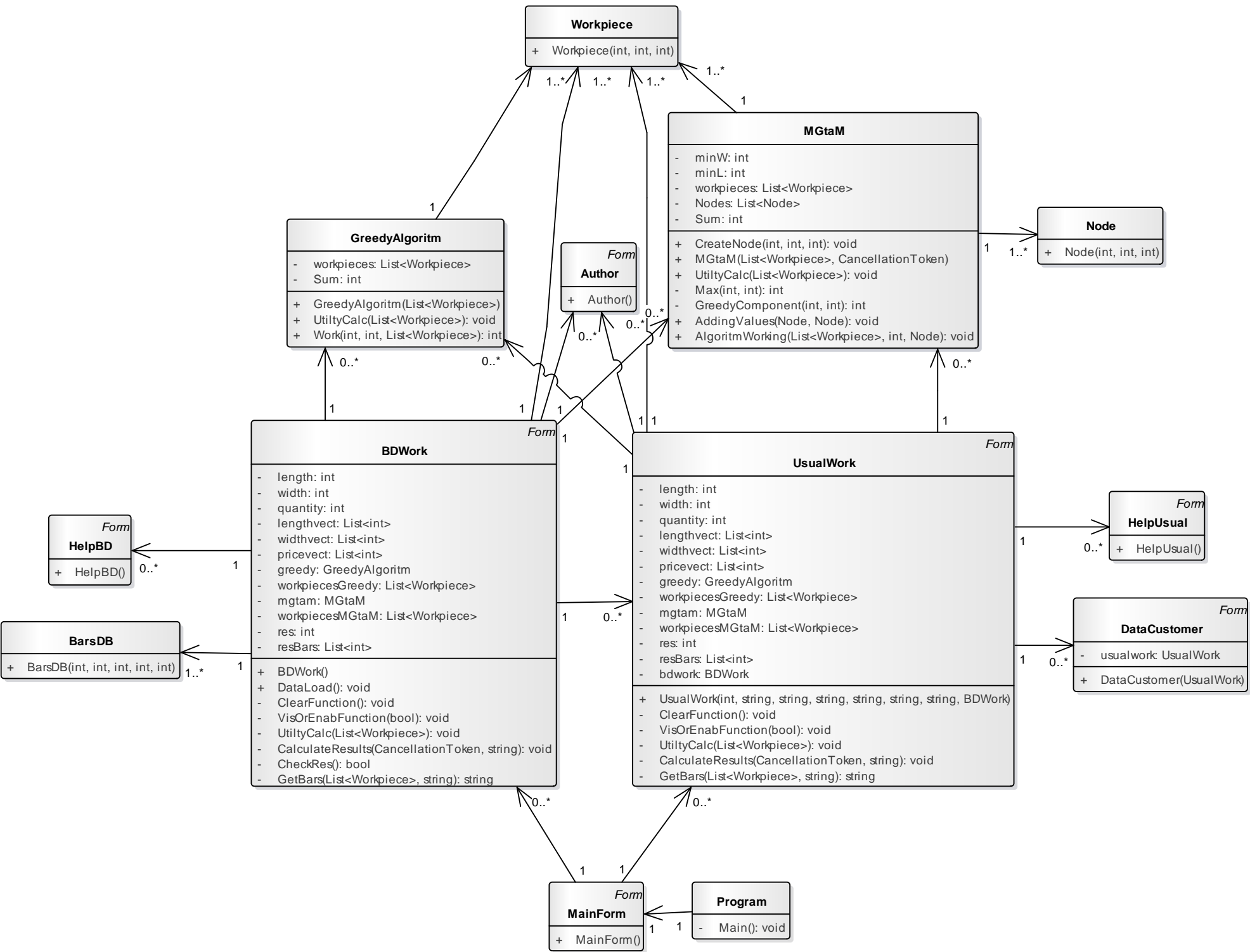


					ДП 6316.02.000 ССВ						
					Схема структурна варіантів використань	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Мосьпак О.І.									
Перевірів		Жураковська О.С.									
						Аркуш 1		Аркушів 1			
Т. кон.					Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63					
Н. кон.		Проскура С.Л.									
Затвердив		Жураковська О.С.									

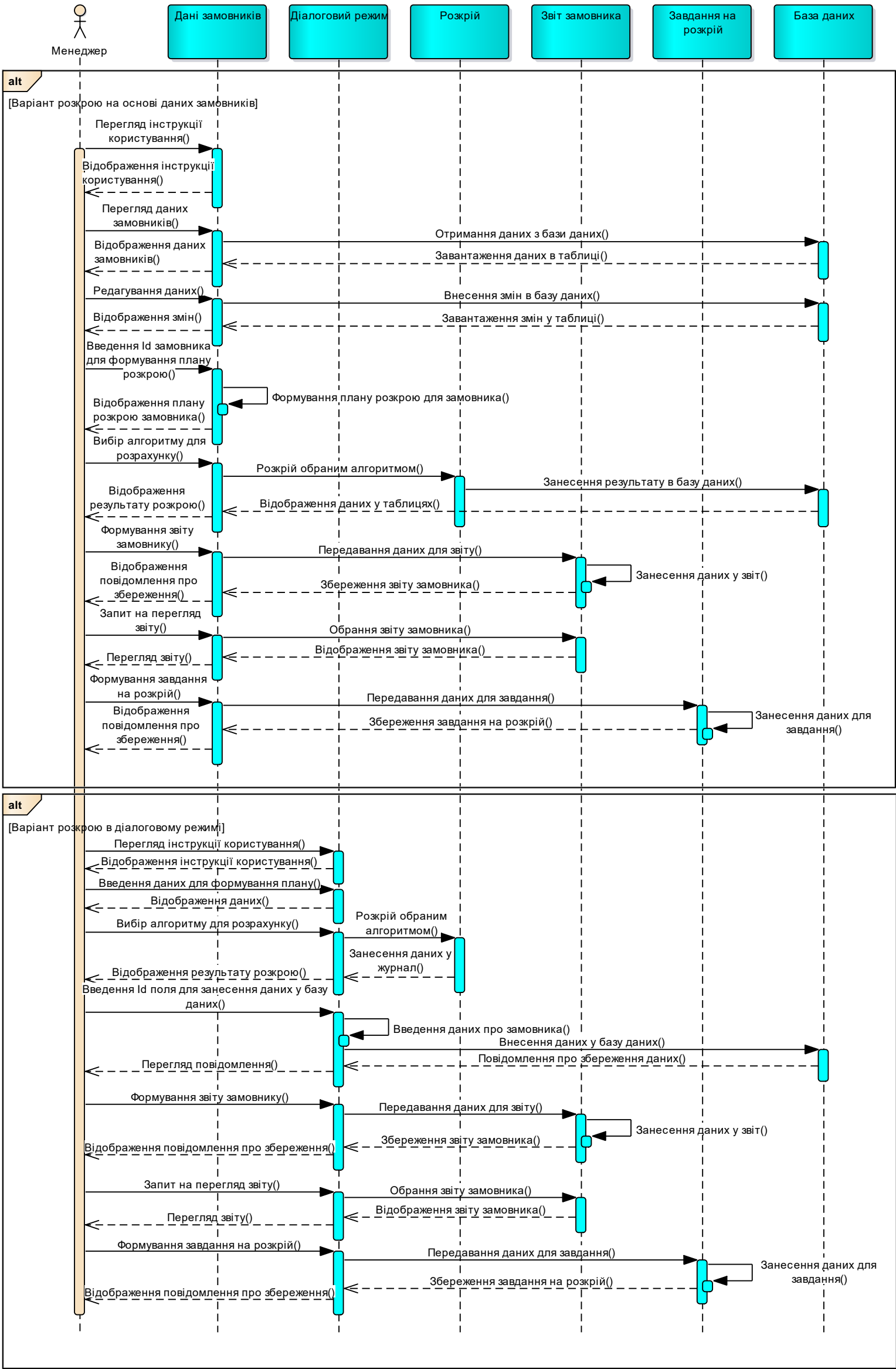




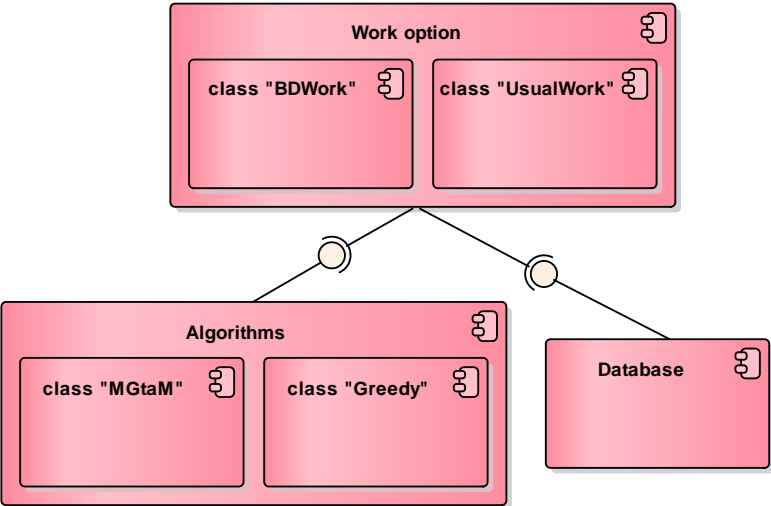
					ДП 6316.04.000 СБД								
Зм.	Арк.	№ документа	Підпис	Дата	Схема бази даних				Літера		Маса	Масштаб	
Розробив	Мосьпак О.І.												
Перевірів	Жураковська О.С.								Аркуш 1		Аркушів 1		
Т. кон.					Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				
Н. кон.	Проскура С.Л.												
Затвердив	Жураковська О.С.												



					ДП 6316.05.000 ССК				
					Схема структурна класів програмного забезпечення				
Зм.	Арк.	№ документа	Підпис	Дата	Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва				
Розробив	Мосьпак О.І.								
Перевірів	Жураковська О.С.								
Т. кон.					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63				
Н. кон.	Проскура С.Л.								
Затвердив	Жураковська О.С.								



					ДП 6316.06.000 ССП						
					Схема структурна послідовності			Лит.		Маса	Масштаб
Зм.	Арк.	№ докум.	Підп.	Дата				Аркуш 1		Аркушів 1	
Розроб.		Мосьяк О.І.									
Перев.		Жураковська О.С.									
Т. Кон.											
Н. Кон.		Проскура С.Л.									
Затв.		Жураковська О.С.									
					Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва			КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63			



					ДП 6316.07.000 ССК						
					Схема структурна компонентів	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Мосьпак О.І.									
Перевірив		Жураковська О.С.									
Т. кон.					Інформаційна система підтримки діяльності розкрийного цеху меблевого виробництва	Аркуш 1		Аркушів 1			
Н. кон.		Проскура С.Л.				КПІ ім. Ігоря Сікорського Каф. АСОІУ					
Затвердив		Жураковська О.С.				Гр. ІС-63					

Work with data of customers

Допомога Про автора

Вибір варіанту розкрою на основі даних замовників

Таблиця "Замовники"

	Id	Name	Surname	Phone	Length_Main	Width_Main	Count_Type_I
▶	6	Margo	Cool	679568421	312	237	3
	9	Nastya	Bus	632322152	380	432	3
	10	Fedor	Car	935558825	362	174	2
	12	Nikita	Last	685462352	300	200	2
	16	Danilo	U	685642535	230	237	3
	18	Oleh	Hall	567833452	312	198	3

Додати Зберегти Видалити

Робота з таблицями

Таблиця "Заготовки"

	Id	Customer_	Bar	Length	Width	Profit
▶	15	6	1	144	34	30
	16	6	2	48	136	87
	17	6	3	97	48	55
	18	9	1	86	89	46
	19	9	2	85	20	29
	20	9	3	25	115	51

Додати Зберегти Видалити

Таблиця "Результати доходу"

	Id	Customer_Id	Algorithm	Total_Income
	23	6	Greedy	522
▶	24	6	MGtaM	660

Введіть Id замовника для формування плану: 6 OK

Вхідні дані

Довжина головної дошки: 312 Роздільник - пробіл Довжини заготовок: 144 48 97

Ширина головної дошки: 237 Ширини заготовок: 34 136 48

Кількість заготовок: 3 Прибутковість заготовок: 30 87 55

Оберіть алгоритм для розрахунку: Метод гілок та меж

Прибуток: 660 Кількість використаних заготовок: 0 0 12

Вибрати звіт для перегляду: Відкрити звіт

Вибрати звіт

Порахувати Зупинити Сформувати звіт Сформувати завдання на розкрій

Робота в діалоговому режимі

Welcome back!

Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва

Оберіть варіант роботи:

Вибір варіанту розкрою на основі даних замовників(з бази даних)

Вибір варіанту розкрою в діалоговому режимі

Вихід

Interactive mode

Допомога Про автора

Вибір варіанту розкрою в діалоговому режимі

Журнал результатів розкрою

	Id	Algorithm	Length_Ma	Width_Main	Count_Type	Vector_Len	Vector_Wid	Vector_Prof	Vector_Res	Income
	1	Greedy	312	198	3	144 48 97	34 136 48	30 87 55	0 6 0	522
▶	2	MGtaM	312	198	3	144 48 97	34 136 48	30 87 55	0 0 12	660

Вхідні дані

Довжина головної дошки: 312 Роздільник - пробіл Довжини заготовок: 144 48 97

Ширина головної дошки: 198 Ширини заготовок: 34 136 48

Кількість заготовок: 3 Прибутковість заготовок: 30 87 55

Введіть Id для занесення даних в базу даних: OK

Оберіть алгоритм для розрахунку: Метод гілок та меж

Прибуток: 660 Кількість використаних заготовок: 0 0 12

Очистити поля

Вибрати звіт для перегляду: Відкрити звіт

Зупинити Порахувати Сформувати звіт Сформувати завдання на розкрій

					ДП 6316.08.000 KE			
Зм.	Арк.	№ документа	Підпис	Дата	Креслення вигляду екранних форм	Літера	Маса	Масштаб
Розробив	Мосьпак О.І.							
Перевірив	Жураковська О.С.				Інформаційна система підтримки діяльності розкрійного цеху меблевого виробництва	Аркуш 1 Аркушів 1		
Т. кон.								
Н. кон.	Проскура С.Л.							
Затвердив	Жураковська О.С.							КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63